THE COMPUTRONICS INC.

EVERYTHING FOR YOUR TRS-80™ • ATARI™ • APPLE™ • PET™ • CP/M™ • XEROX™ • IBM™ • OSBORNE™ • • • • KAYPRO • •

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. * ATARI is a trademark of Atari Inc. * APPLE is a trademark of Apple Corp. * PET is a trademark of Commodore * CP/M is a trademark of Digital Research * XEROX is a trademark of Xerox Corp. * IBM is a trademark of IBM Corp. * OSBORNE is a trademark of Osborne Corp.

BUSINESS PAC 100

100 Ready-To-Run Business Programs

(ON CASSETTE OR DISKETTE).....Includes 128 Page Users Manual.....
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

NAME	DESCRIPTION
1 RULE78	Interest Apportionment by Rule of the 78's
2 ANNU1	Annuity computation program
3 DATE	Time between dates
4 DAYYEAR	Day of year a particular date falls on
[illegible]EASEINT	Interest rate on lease
[illegible]EAKEVN	Breakeven analysis
[illegible]PRSL	Straightline depreciation
8 DEPRSY	Sum of the digits depreciation
9 DEPRDB	Declining balance depreciation
10 DEPRDDB	Double declining balance depreciation
11 CADEP	Cash flow vs. depreciation tables
12 CHECK2	Prints NEBS checks along with daily register
13 CHECKBK1	Checkbook maintenance program
14 MORTGAGE.A	Mortgage amortization table
15 MULTMON	Computes time needed for money to double, triple, etc.
16 SALVAGE	Determines salvage value of an investment
17 RRVARIN	Rate of return on investment with variable inflows
18 RRCONST	Rate of return on investment with constant inflows
19 EFFECT	Effective interest rate of a loan
20 FVAL	Future value of an investment (compound interest)
21 PVAL	Present value of a future amount
22 LOANPAY	Amount of payment on a loan
23 PEGWITH	Equal withdrawals from investment to leave 0 over
24 SIMPDISK	Simple discount analysis
25 DATEVAL	Equivalent & nonequivalent dated values for oblig.
26 ANNUDEF	Present value of deferred annuities
27 MARKUP	% Markup analysis for items
28 SINKFUND	Sinking fund amortization program
29 BONDVAL	Value of a bond
30 DEPLETE	Depletion analysis
31 BLACKSH	Black Scholes options analysis
32 STOCVAL1	Expected return on stock via discounts dividends
33 WARVAL	Value of a warrant
34 BONDVAL2	Value of a bond
35 EPSEST	Estimate of future earnings per share for company
36 BETAALPH	Computes alpha and beta variables for stock
37 SHARPE1	Portfolio selection model-i.e. what stocks to hold
38 OPTWRITE	Option writing computations
39 RTVAL	Value of a right
40 EXPVAL	Expected value analysis
41 BAYES	Bayesian decisions
42 VALPRINF	Value of perfect information
43 VALADINF	Value of additional information
44 UTILITY	Derives utility function
45 SIMPLEX	Linear programming solution by simplex method
46 TRANS	Transportation method for linear programming
47 EOQ	Economic order quantity inventory model
[illegible]QUEUE1	Single server queueing (waiting line) model
[illegible]VP	Cost-volume-profit analysis
[illegible]ONDPROF	Conditional profit tables
[illegible]PTLOSS	Opportunity loss tables
52 FQUOQ	Fixed quantity economic order quantity model
53 FQEOWSH	As above but with shortages permitted
54 FQEOQPB	As above but with quantity price breaks
55 QUEUECB	Cost-benefit waiting line analysis
56 NCFANAL	Net cash flow analysis for simple investment
57 PROFIND	Profitability index of a project
58 CAPI	Cap. Asset Pr. Model analysis of project
59 WACC	Weighted average cost of capital
60 COMPBAL	True rate on loan with compensating bal. required
61 DISCBAL	True rate on discounted loan
62 MERGANAL	Merger analysis computations
63 FINRAT	Financial ratios for a firm
64 NPV	Net present value of project
65 PRINDLAS	Laspeyres price index
66 PRINDPA	Paasche price index
67 SEASIND	Constructs seasonal quantity indices for company
68 TIMETR	Time series analysis linear trend
69 TIMEMOV	Time series analysis moving average trend
70 FUTPRINF	Future price estimation with inflation
71 MAILPAC	Mailing list system
72 LETWRT	Letter writing system links with MAILPAC
73 SORT3	Sorts list of names
74 LABEL1	Shipping label maker
75 LABEL2	Name label maker
76 BUSBUD	DOME business bookkeeping system
77 TIMECLCK	Computes weeks total hours from timeclock info.
78 ACCTPAY	In memory accounts payable system-storage permitted
79 INVOICE	Generate invoice on screen and print on printer
80 INVENT2	In memory inventory control system
81 TELDIR	Computerized telephone directory
82 TIMUSAN	Time use analysis
83 ASSIGN	Use of assignment algorithm for optimal job assign.
84 ACCTREC	In memory accounts receivable system-storage ok
85 TERMSPAY	Compares 3 methods of repayment of loans
86 PAYNET	Computes gross pay required for given net
87 SELLPR	Computes selling price for given after tax amount
88 ARBCOMP	Arbitrage computations
89 DEPRSF	Sinking fund depreciation
90 UPSZONE	Finds UPS zones from zip code
91 ENVELOPE	Types envelope including return address
92 AUTOEXP	Automobile expense analysis
93 INSFILE	Insurance policy file
94 PAYROLL2	In memory payroll system
95 DILANAL	Dilution analysis
96 LOANAFFD	Loan amount a borrower can afford
97 RENTPRCH	Purchase price for rental property
98 SALELEAS	Sale leaseback analysis
99 RRCONVBD	Investor's rate of return on convertable bond
100 PORTVAL9	Stock market portfolio storage-valuation program

*** IMPORTANT REGISTRATION INFORMATION ***

We have done our best to insure that there are no errors in your H & E COMPUTRONICS INC. software package. By providing our customers with a FREE periodic software update newsletter, we have become the first software house to insure that software buyers become aware of the latest software changes and corrections. To receive our FREE software update publication, please read our registration information below and return the enclosed form.

1. Although H & E Computronics, Inc., has done its best to insure that this software package does not contain any errors, it is the user's responsibility to thoroughly test out the software package.

2. H & E Computronics, Inc. does not take any responsibility for any errors that may appear in this software. The end user is responsible for detecting and correcting any errors that may appear in the software.

3. H & E Computronics, Inc. does offer its customers a 30-day money back guarantee on its software packages. Complete information related to our return policy is included on the back of every invoice that accompanies every software package purchased from H & E Computronics, Inc. If you purchased your software package from a vendor other than H & E Computronics, Inc., the user must consult that vendor for its return policy (if any).

4. Return this original Registration Form (copies are not acceptable). When we receive your form, you will be placed on our software newsletter update list. H & E Computronics, Inc. will inform all software users of any changes, enhancements or error corrections that we may become aware of. This service will be provided FREE of charge for a period of one year from the registration date (charges, if any, after the end of the one year period of time have not yet been determined).

5. DO NOT REMOVE THIS REGISTRATION FORM UNLESS YOU ARE SURE THAT YOU ARE KEEPING THIS SOFTWARE PACKAGE. ONCE THIS REGISTRATION FORM IS REMOVED, THIS SOFTWARE PACKAGE MAY NOT BE RETURNED FOR ANY REASON.

6. In the event that the diskette(s) in your software package does (do) not work properly (due to faulty diskettes, programming errors, magnetic fields, incorrect software version, etc.), the user must return the diskette(s) to H & E Computronics, Inc. for testing and proper evaluation. As an alternative, H & E Computronics, Inc. will sell the user (an) additional diskette(s) at $5 per diskette (plus $3 postage and handling per order). This offer is only valid within 30-days of the purchase of your software.

7. All registered owners are entitled to update their software package at any time for $25 (plus $3 postage and handling). This update fee includes a new manual and (a) new diskette(s). This policy is especially helpful to software owners without any programming experience. Non-experienced owners will be able to purchase updated software at a nominal fee (instead of making any changes that may appear in the update newsletter).

8. The cost of computer programmers is very high. H & E Computronics, Inc. is not responsible for providing any on-site or telephone service to software owners. Because we take pride in our software packages and are interested in insuring that all H & E Computronics, Inc. software buyers get the full value of the software that they have purchased, we do provide software purchasers with as much telephone service as possible. If you call H & E Computronics, Inc. between the hours of 9 AM and 5 PM (New York time), we will connect you with a programmer that is knowledgeable about the specific software package that you purchased. This programmer will try to answer any questions that you may have,

H & E COMPUTRONICS INC.

MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100 INTRODUCTION and INSTRUCTIONS

Welcome to the world of fine diversified computing. Your **H & E COMPUTRONICS BUSINESS PAC 100** is a carefully chosen set of 100 pre-recorded programs designed for utility, education and their business applicability. There is a complete set of **financial computation programs** which will permit the user to do everything from figuring proper payments on a loan to comparing cash flow and amount of depreciaiton for a small business. Included are an **inventory control program**, a **business bookkeeping program**, a **checkbook maintenance program**, a **mailing list program** linked to a text editor for repetitive letter writing and many more. Programs aimed towards the stock market are included as well as those aiding in the use of modern management science techniques. There are balance forward A/R and A/P systems, a payroll system, etc. Every small businessman should find several useful programs in the **BUSINESS PAC 100.**

Most of the programs found on the **BUSINESS PAC** are self documenting. This means that they can be run and (hopefully) understood without any instructions not found in the programs themselves. However, the **BUSINESS PAC** comes with written documentation for every program. Most programs have one page of written documentation. Several, such as the inventory control and mailing list programs require two pages. ACCTREC and ACCTPAY share 4 pages. With the written documentation and the in-program documentation, programs should be relatively simple to run. Of course, if one wishes to run an inventory control program involving the economic order quantity it would help to understand something of the concept. The documentation does try to help somewhat with this understanding, but one page is not alot.

The programs which involve external storage may all be used on cassette and diskette. If one upgrades from a cassette to a diskette it will not even be necessary to recopy files. The cassette file may be loaded into memory and then saved on diskette in a single run of the program. Dumping a file from diskette to a cheaper cassette for archiving purposes is also supported.

You are probably anxious to try out your **BUSINESS PAC 100** so we'll say something about loading and how to read the enclosed CONTENTS AND LOCATION sheet. Those receiving this package on cassette will receive 5 cassettes duplicated on the finest equipment by a major recording firm. There should not be any problem loading the individual programs. However, it may be difficult to load programs in even this case. Sometimes it is necessary to save a program on the recorder which will be used to load. So try to find the proper recorder volume, CLOAD the appropriate program and if there is any problem in loading CSAVE the program on the recorder in use with the computer. Of course, if the diskette version of the **BUSINESS PAC** is in use there should be no such problem. The program LOADS up with the LOAD"NAME" command. The diskette version does come on 5 formatted diskettes without operating systems. With a multi-drive system this presents no problem. In

case a single drive system is in use the user has several options. He may purchase a RS operating system or send one of his own to H & E COMPUTRONICS in order to get diskettes with operating systems. He may also purchase the NEWDOS-80 operating system which permits copying of diskettes when only one drive is available and where one diskette may not have an operating system. At present, using the RS operating system with a one drive system is impossible unless an operating system is always on the diskette in that drive.

Every program should load and run in a 16K LEVEL II TRS-80 w/o diskette or a 32K LEVEL II with diskette. If more memory is available then several programs may be dimensioned to handle more data than on the distribution system. This is generally noted in the documentation.

Reading the enclosed CONTENTS AND LOCATION directory should be extremely simple. For each program is found the program number, the disk file name (LOAD "FILENAME" from disk BASIC to load), the disk and tape number on which the program may be found(1 to 5 . There are about 20 programs on each.), the cassette name of the program along with position on the tape (e.g. ACCTPAY has cassette name A and is loaded from cassette by pressing CLOAD#-1,"A" and <ENTER>. Also, since A is the 1st letter of the alphabet we know that BARGRAPH is the 1st program on the tape), and a brief description of each program. The following is brief section from the CONTENTS AND LOCATION.

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
1	ACCTPAY	5	A	ACCOUNTS PAYABLE SYST
2	ACCTREC	5	B	ACCOUNTS RECEIVABLE
3	ANNU1	5	C	ANNUITIES COMPUTATION

Program number 3 is found on DISK 5 or TAPE 5. From disk it is loaded by placing the diskette into drive 1,(or drive 0 if purchased with an operating system), load DISK BASIC and then entering LOAD"ANNU1" <ENTER>. From tape it is loaded by placing Tape 5 into the cassette recorder, putting the recorder into the PLAY mode, and press CLOAD#-1,"C" <ENTER>. Under the DESCRIPTION column we see that this is an annuities computation program.

THERE ARE NO SYSTEM PROGRAMS IN BUSINESS PAC.ALL PROGRAMS ARE EITHER LOADED FROM DISK BASIC OR CLOADED FROM LEVEL II BASIC.

To determine position on the tape vis a vis the tape counter we recommend carefully loading each program while copying down the appropriate counter information, i.e. start and finish (where the blinking light starts & where the recorder stops.). Then on subsequent loads the user need only fast forward to just before the beginning number and CLOAD.

Several of the programs require use of a line printer to be used intelligently. As examples we have the label making programs. Others give the option of printed or CRT output. If printed output is desired all that is necessary is to answer the question relating to printing appropriately.

All data files are sequential files. This choice of file type was made to keep all files compatible whether they reside on diskette or cassette. Sequential files are very easy to deal with and even novices should have very little trouble in modifying the various file handling programs for individual needs.

For those wishing to place any of these programs into production in a business environment note that most computations are done using single precision arithmetic. This is mainly due to the fact that standard LEVEL II and DISK BASIC do not support double precision for any of the complex mathematical functions like the exponential function which is used in almost all financial calculations. The results obtained in the sample runs of the various programs were excellent, even without double precision. However, the user must note that dollar amounts of slightly more than $10,000 may be off by a penny or two with greater errors in the $100,000 range. If this deviation is important in the particular computation in question and if the only operations used are +, -, x, and / then double precision variables may be defined. Remember, if a single precision variable is used anywhere then subsequent double precision results may print out to 16 places but will only be accurate to about 6 places.

We hope that the **BUSINESS PAC 100 makes thousands of users happy like its predescessor the** MASTER PAC 100. **It will probably take weeks to explore this diverse cornucopia of fine programs. The philosophy used throughout has been to illustrate many uses for many business people.**

HAPPY COMPUTING

Peter Shenkin, Ph.D.
October, 1980

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100-CONTENTS AND LOCATION

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
1	ACCTPAY	5	A	IN MEMORY A/P SYSTEM
2	ACCTREC	5	B	IN MEMORY A/R SYSTEM
3	ANNU1	5	C	ANNUITY COMPUTATION PROG.
4	ANNUDEF	3	A	DEFERRED ANNUITY COMP.
5	ARBCOMP	1	A	ARBITRAGE COMPUTATIONS
6	ASSIGN	5	D	ASSIGNMENT ALGORITHM
7	AUTOEXP	2	A	AUTOMOBILE EXPENSE ALAL.
8	BAYES	3	B	BAYESIAN DECISIONS
9	BETAALPH	2	B	ALPHA & BETA STOCK CALC.
10	BLACKSH	5	E	BLACK-SCHOLES OPTION ANAL
11	BONDVAL	2	C	VALUE OF A BOND
12	BONDVAL2	5	F	VALUE OF A BOND
13	BREAKEVN	4	A	BREAKEVEN ANALYSIS
14	BUSBUD	5	G	BUSINESS BOOKKEEPING SYST
14	CAP1	1	B	CAPITAL ASSET PRICING MOD
16	CHECK2	5	H	PRINTS CHECKS + REGISTER
17	CHECKBK1	5	I	CHECKBOOK MAINTENANCE
18	COMPBAL	1	C	COMPENSATING BAL. LOAN
19	CONDPROF	3	C	CONDITIONAL PROFIT TABLE
20	CVP	1	D	COST-VOLUME-PROFIT ANAL
21	DATE	5	J	TIME BETWEEN DATES
22	DATEVAL	3	D	DATED VALUES FOR OBLIGAT.
23	DAYYEAR	1	E	DAY OF YEAR FOR DATE
24	DEPLETE	1	F	DEPLETION ANALYSIS
25	DEPRDB	5	K	DECLINING BALANCE DEPR.
26	DEPRDDB	1	G	DOUBLE DECLIN BAL DEPR.
27	DEPRSF	1	H	SINKING FUND DEPREC.
28	DEPRSL	5	L	STRAIGHT LINE DEPRECIAT.
29	DEPRSY	1	I	SUM OF YRS DIGITS DEPR.
30	DILANAL	1	J	DILUTION ANALYSIS
31	DISCBAL	1	K	RATE ON DISCOUNTED LOAN
32	EFFECT	4	B	EFFECTIVE INTEREST RATE
33	ENVELOPE	1	L	PRINTS ENVELOPE-HARD COPY
34	EOQ	3	E	EOQ INVENTORY MODEL

H&E COMPUTRONICS INC.

MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100-CONTENTS AND LOCATION

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
35	EPSEST	2	D	EARNINGS PER SHARE EST.
36	EXPVAL	3	F	EXPECTED VALUE ANALYSIS
37	FINRAT	1	M	FINANCIAL RATIOS FOR FIRM
38	FQEOQ	3	G	FIXED QUANTITY EOQ
39	FQEOQPB	4	C	ORDER WITH QUANT PR.BREAK.
40	FQEOWSH	3	H	ORDER WITH SHORTAGES OK
41	FUPRINF	2	E	PRICE ESTIM. WITH INFLAT.
42	FVAL	4	D	FUTURE VALUE OF INVEST.
43	INSFILE	2	F	INSURANCE POLICY FILE
44	INVENT2	5	M	IN MEM. INVENTORY CONTROL
45	INVOICE	4	E	INVOICING PROGRAM
46	LABEL1	2	G	SHIPPING LABEL MAKER
47	LABEL2	2	H	NAME LABEL MAKER
48	LEASEINT	4	F	INTEREST RATE ON LEASE
49	LETWRT	4	G	TEXT EDITOR-MAILPAC LINK
50	LOANAFFD	1	N	LOAN AMT. YOU CAN AFFORD
51	LOANPAY	3	I	PAYMENT ON LOAN
52	MAILPAC	4	H	IN MEM. MAILING LIST PRG.
53	MARKUP	1	O	% MARKUP ANALYSIS
54	MERGANAL	1	P	MERGER ANALYSIS
55	MORTGAGE/A	4	I	MORTGAGE AMORTIZATION
56	MULTMON	3	J	TIME TO MULTIPLY MONEY
57	NCFANAL	4	J	NET CASH FLOW ANALYSIS
58	NPV	1	Q	NET PRESENT VALUE
59	OPTLOSS	3	K	OPPORTUNITY LOSS TABLE
60	OPTWRITE	3	L	OPTION WRITING COMPUTAT.
61	PAYNET	2	I	GROSS PAY FOR GIVEN NET
62	PAYROLL2	4	K	IN MEMORY PAYROLL SYSTEM
63	PORTVAL9	4	L	PORTFOLIO STORAGE-EVALUAT
64	PRINDLAS	2	J	LASPEYRES PRICE INDEX
65	PRINDPA	1	R	PAASCHE PRICE INDEX
66	PROFIND	2	K	PROFITABILITY INDEX
67	PVAL	4	M	PRESENT VALUE OF AMOUNT

BUSINESS PAC 100-CONTENTS AND LOCATION

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
68	QUEUE1	4	N	1 SERVER WAITING LINES
69	QUEUECB	3	M	COST BENEFIT QUEUING THR.
70	REGWITH	4	O	REGULAR WITHDRAWALS
71	RENTPRCH	1	S	PURCH.PR. FOR RENTAL PROP
72	RRCONST	4	P	RATE OF RETURN-CONST INFL
73	RRCONVBD	1	T	CONV.BOND-RATE OF RETURN
74	RRVARIN	5	N	RATE OF RET-VARIABLE INFL
75	RTVAL	3	N	VALUE OF A RIGHT
76	RULE78	1	U	RULE OF 78'S INT CALC
77	SALELEAS	5	O	SALE-LEASEBACK ANALYSIS
78	SALVAGE	5	P	SALVAGE VALUE
79	SEASIND	2	L	SEASONAL INDEX COMPUT.
80	SELLPR	2	M	SELL PRICE-AFTER TAX
81	SHARPE1	2	N	PORTFOLIO SELECTION PROG.
82	SIMPDISK	4	Q	SIMPLE DISCOUNT ANALYSIS
83	SIMPLEX	2	O	LINEAR PROGRAMMING
84	SINKFUND	3	O	SINKING FUND AMORTIZATION
85	SORT3	3	P	ALPHABETIC SORTING
86	STOCVAL1	4	R	STOCK VALUATION
87	TAXDEP	5	Q	CASH FLOW VS. DEPRECIAT.
88	TELDIR	2	P	TELEPHONE DIRECTORY
89	TERMSPAY	4	S	COMPARES LOAN REPAY TYPES
90	TIMECLCK	3	Q	COMP. HRS FROM TIMECLOCK
91	TIMEMOV	4	T	TIME SERIES-MOVING AVG.
92	TIMETR	4	U	TIME SERIES-LINEAR TREND
93	TIMUSAN	3	R	TIME USE ANALYSIS
94	TRANS	4	V	TRANSPORTATION PROBLEM
95	UPSZONE	4	W	FIND UPS ZONE FROM ZIP
96	UTILITY	3	S	DERIVES UTILITY FUNCTION
97	VALADINF	3	T	VALUE OF ADDITIONAL INFO
98	VALPRINF	2	Q	VALUE OF PERFECT INFO
99	WACC	5	R	WGHT AVG COST OF CAPITAL
100	WARVAL	3	U	VALUE OF A WARRANT

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100-CONTENTS OF VOLUME 1

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
5	ARBCOMP	1	A	ARBITRAGE COMPUTATIONS
14	CAP1	1	B	CAPITAL ASSET PRICING MOD
18	COMPBAL	1	C	COMPENSATING BAL. LOAN
20	CVP	1	D	COST-VOLUME-PROFIT ANAL
23	DAYYEAR	1	E	DAY OF YEAR FOR DATE
24	DEPLETE	1	F	DEPLETION ANALYSIS
26	DEPRDDB	1	G	DOUBLE DECLIN BAL DEPR.
27	DEPRSF	1	H	SINKING FUND DEPREC.
29	DEPRSY	1	I	SUM OF YRS DIGITS DEPR.
30	DILANAL	1	J	DILUTION ANALYSIS
31	DISCBAL	1	K	RATE ON DISCOUNTED LOAN
33	ENVELOPE	1	L	PRINTS ENVELOPE-HARD COPY
37	FINRAT	1	M	FINANCIAL RATIOS FOR FIRM
50	LOANAFFD	1	N	LOAN AMT. YOU CAN AFFORD
53	MARKUP	1	O	% MARKUP ANALYSIS
54	MERGANAL	1	P	MERGER ANALYSIS
58	NPV	1	Q	NET PRESENT VALUE
65	PRINDPA	1	R	PAASCHE PRICE INDEX
71	RENTPRCH	1	S	PURCH.PR. FOR RENTAL PROP
73	RRCONVBD	1	T	CONV.BOND-RATE OF RETURN
76	RULE78	1	U	RULE OF 78'S INT CALC

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100-CONTENTS OF VOLUME 2

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
7	AUTOEXP	2	A	AUTOMOBILE EXPENSE ALAL.
9	BETAALPH	2	B	ALPHA & BETA STOCK CALC.
11	BONDVAL	2	C	VALUE OF A BOND
35	EPSEST	2	D	EARNINGS PER SHARE EST.
41	FUPRINF	2	E	PRICE ESTIM. WITH INFLAT.
43	INSFILE	2	F	INSURANCE POLICY FILE
46	LABEL1	2	G	SHIPPING LABEL MAKER
47	LABEL2	2	H	NAME LABEL MAKER
61	PAYNET	2	I	GROSS PAY FOR GIVEN NET
64	PRINDLAS	2	J	LASPEYRES PRICE INDEX
66	PROFIND	2	K	PROFITABILITY INDEX
79	SEASIND	2	L	SEASONAL INDEX COMPUT.
80	SELLPR	2	M	SELL PRICE-AFTER TAX
81	SHARPE1	2	N	PORTFOLIO SELECTION PROG.
83	SIMPLEX	2	O	LINEAR PROGRAMMING
88	TELDIR	2	P	TELEPHONE DIRECTORY
98	VALPRINF	2	Q	VALUE OF PERFECT INFO

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100-CONTENTS OF VOLUME 3

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
4	ANNUDEF	3	A	DEFERRED ANNUITY COMP.
8	BAYES	3	B	BAYESIAN DECISIONS
19	CONDPROF	3	C	CONDITIONAL PROFIT TABLE
22	DATEVAL	3	D	DATED VALUES FOR OBLIGAT.
34	EOQ	3	E	EOQ INVENTORY MODEL
36	EXPVAL	3	F	EXPECTED VALUE ANALYSIS
38	FQEOQ	3	G	FIXED QUANTITY EOQ
40	FQEOWSH	3	H	ORDER WITH SHORTAGES OK
51	LOANPAY	3	I	PAYMENT ON LOAN
56	MULTMON	3	J	TIME TO MULTIPLY MONEY
59	OPTLOSS	3	K	OPPORTUNITY LOSS TABLE
60	OPTWRITE	3	L	OPTION WRITING COMPUTAT.
69	QUEUECB	3	M	COST BENEFIT QUEUING THR.
75	RTVAL	3	N	VALUE OF A RIGHT.
84	SINKFUND	3	O	SINKING FUND AMORTIZATION
85	SORT3	3	P	ALPHABETIC SORTING
90	TIMECLCK	3	Q	COMP. HRS FROM TIMECLOCK
93	TIMUSAN	3	R	TIME USE ANALYSIS
96	UTILITY	3	S	DERIVES UTILITY FUNCTION
97	VALADINF	3	T	VALUE OF ADDITIONAL INFO
100	WARVAL	3	U	VALUE OF A WARRANT

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100-CONTENTS OF VOLUME 4

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
13	BREAKEVN	4	A	BREAKEVEN ANALYSIS
32	EFFECT	4	B	EFFECTIVE INTEREST RATE
39	FQEOQPB	4	C	ORDER WITH QUANT PR.BREAK
42	FVAL	4	D	FUTURE VALUE OF INVEST.
45	INVOICE	4	E	INVOICING PROGRAM
48	LEASEINT	4	F	INTEREST RATE ON LEASE
49	LETWRT	4	G	TEXT EDITOR-MAILPAC LINK
52	MAILPAC	4	H	IN MEM. MAILING LIST PRG.
55	MORTGAGE/A	4	I	MORTGAGE AMORTIZATION
57	NCFANAL	4	J	NET CASH FLOW ANALYSIS
62	PAYROLL2	4	K	IN MEMORY PAYROLL SYSTEM
63	PORTVAL9	4	L	PORTFOLIO STORAGE-EVALUAT
67	PVAL	4	M	PRESENT VALUE OF AMOUNT
68	QUEUE1	4	N	1 SERVER WAITING LINES
70	REGWITH	4	O	REGULAR WITHDRAWALS
72	RRCONST	4	P	RATE OF RETURN-CONST INFL
82	SIMPDISK	4	Q	SIMPLE DISCOUNT ANALYSIS
86	STOCVAL1	4	R	STOCK VALUATION
89	TERMSPAY	4	S	COMPARES LOAN REPAY TYPES
91	TIMEMOV	4	T	TIME SERIES-MOVING AVG.
92	TIMETR	4	U	TIME SERIES-LINEAR TREND
94	TRANS	4	V	TRANSPORTATION PROBLEM
95	UPSZONE	4	W	FIND UPS ZONE FROM ZIP

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BUSINESS PAC 100-CONTENTS OF VOLUME 5

NO.	DISK NAME	DISK/TAPE	TAPE NAME	DESCRIPTION
1	ACCTPAY	5	A	IN MEMORY A/P SYSTEM
2	ACCTREC	5	B	IN MEMORY A/R SYSTEM
3	ANNU1	5	C	ANNUITY COMPUTATION PROG.
6	ASSIGN	5	D	ASSIGNMENT ALGORITHM
10	BLACKSH	5	E	BLACK-SCHOLES OPTION ANAL
12	BONDVAL2	5	F	VALUE OF A BOND
14	BUSBUD	5	G	BUSINESS BOOKKEEPING SYST
16	CHECK2	5	H	PRINTS CHECKS + REGISTER
17	CHECKBK1	5	I	CHECKBOOK MAINTENANCE
21	DATE	5	J	TIME BETWEEN DATES
25	DEPRDB	5	K	DECLINING BALANCE DEPR.
28	DEPRSL	5	L	STRAIGHT LINE DEPRECIAT.
44	INVENT2	5	M	IN MEM. INVENTORY CONTROL
74	RRVARIN	5	N	RATE OF RET-VARIABLE INFL
77	SALELEAS	5	O	SALE-LEASEBACK ANALYSIS
78	SALVAGE	5	P	SALVAGE VALUE
87	TAXDEP	5	Q	CASH FLOW VS. DEPRECIAT.
99	WACC	5	R	WGHT AVG COST OF CAPITAL

BUSINESS PAC 100 ADDENDUM

A NOTE ABOUT YOUR MANUAL

The **BUSINESS PAC 100** manual was originally written for the TRS-80 MODEL I WITH CASSETTE. The program loading instructions are accurate ONLY on the MODEL I WITH CASSETTE. If you have another computer, please read the instructions in this addendum before calling H & E COMPUTRONICS for technical assistance.

NOTE: Throughout the documentation <ENTER> or <RETURN> denote push the ENTER key or RETURN key on the TRS-80 and APPLE II, respectively. All other references inside the signs [] should be directly typed on the keyboard.

REMEMBER, WHEN YOU SEE []'S IN THIS MANUAL, DO NOT TYPE THE BRACKETS.

TYPE ONLY WHAT YOU SEE INSIDE THE BRACKETS!!!!

Also, please remember that the disk directories in this manual are accurate only for the TRS-80 MODEL I. If you own another computer, you may have received less than 5 disks. See the instructions in this addendum to find out how many disks you were supposed to receive and how to get a 'directory' of what programs are on each disk.

* * * IMPORTANT NOTICE * * *

1. IN THE EVENT THAT THE DISKETTES CONTAINED IN YOUR SOFTWARE PACKAGE DO NOT WORK PROPERLY (DUE TO FAULTY DISKETTES, PROGRAMMING ERRORS, MAGNETIC FIELDS, ETC.), THE USER MUST RETURN THE DISKETTES TO H & E COMPUTRONICS, INC. AS AN ALTERNATIVE, H & E COMPUTRONICS, INC. WILL SELL THE USER ONE ADDITIONAL SET OF DISKETTES AT $5 PER DISKETTE (PLUS $3 POSTAGE AND HANDLING PER ORDER).

2. THERE ARE NO KNOWN PROGRAMMING ERRORS CONTAINED ON THE ENCLOSED DISKETTES. THE USER MAY UPDATE HIS DISKETTES AT ANY TIME BY FOLLOWING THE PROCEDURE IN # 1 ABOVE. IF THE USER SHOULD FIND AN ERROR IN ANY OF THE PROGRAMS, OUR PROGRAMMING STAFF WILL CORRECT THAT ERROR (AT OUR DISCRETION). THE ONLY CHARGE TO THE USER IS THE STANDARD DISKETTE CHARGE AS IN # 1 ABOVE.

COMMODORE/PET OWNERS

There are two versions of the PET/COMMODORE BUSINESS PAC 100. You have received either three diskettes (2040, 3040, and 4040 drives) or one diskette (8050 drive). If you have received a version of the BUSINESS PAC which is not designed for your disk drive, please return your disk(s) (not your manual) to Computronics with a brief note requesting the alternate version of the PET/CBM BUSINESS PAC.

In order to determine which programs are located on each disk, simply do the following:

1) After first making sure that there are no diskettes in either disk drive, turn on your disk drive; then turn on your computer.

2) Insert a BUSINESS PAC diskette into drive 1.

3) Type [DIRECTORY D1] <RETURN> (Do not type []'s)

4) Directories of the other diskettes may be obtained by switching disks and repeating step 3.

5) If you desire a printed copy of the disk DIRECTORY, type [OPEN 4,4:CMD4] <RETURN>. You may then repeat step 3 with each disk. [PRINT#4:CLOSE4] <RETURN> will cancel line printer output.

RUNNING THE PROGRAMS

In order to run a BUSINESS PAC program, do the following:

1) While the disk drive is empty, turn on your disk drive; then turn on your computer.

2) If you are operating a computer with an 80-column screen, you must first load in a program called '80 TO 40 COLUMN' and then run the program. Your screen will initially be covered with many different letters and graphics symbols. Once the program is run it will remain in memory until the computer is shut off. Any program run after loading this program will be executed with a 40-column screen format. IT is essential that owners of 80-column screens run this program. Printer operations will still be performed with the normal number of columns.

3) Type [LOAD"Dn:fn",8] <RETURN>, where n represents the drive number and fn represents the name of the program you wish to run.
   Ex. - [LOAD"D1:ACCTREC",8] <RETURN>.

   (NOTE: The drive on the right is number 0; the drive on the left is number 1.)

4) Type [RUN] <RETURN> and your program will run.

5) Additional programs may be run by simply inserting the appropriate disk and repeating step 4.

6) In order to end a program, press the RUN/STOP key. (Experienced programmers will note that depressing the RETURN key will not terminate the program while inputting data, as happens often with programs written in BASIC for PET/CBM computers.)

A NOTE ABOUT THE PET/CBM VERSION

1) The screen formatting depicted in the manual has, in some cases, been altered in order to accommodate that PET/CBM's 40-character CRT. Some charts have been compressed and inputs may now be prompted in a slightly different manner. These changes were done strictly for aesthetic purposes. They will not alter the performance or accuracy of your BUSINESS PAC 100 programs.

2) Some calculations on the TRS-80 programs are performed with double-precision accuracy. The PET/CBM computers do not support this feature; therefore, some calculations may now be a few cents off from the documented output.

ACCTREC AND ACCTPAY

ACCTREC is a balance forward accounts receivable system. In a balance forward system payments are not credited towards individual invoices or bills but only to the total owed. Invoices or individual bills need not be carried month after month. Balance forward A/R systems are typically used by professionals such as doctors, dentists, accountants, etc. and by charge account companies. In these cases, at least after the first month, the buyer does not pay off individual charges but an amount credited to the entire account. Many businesses use an invoice type A/R system. In this type of system payments are credited towards individual invoices. In an invoice oriented system it is necessary to keep track of each invoice until paid. This is not necessary in a balance forward system. In the particular system used by this program the individual invoices are kept on file until the end of the month in which they were generated. It is at this time that the invoices are posted to the customer accounts and then purged from the system. It is advised that a printed invoice list be generated before this time.

ACCTPAY is a balance forward type accounts payable system and is almost identical to **ACCTREC**. In fact, the only real differences in the programs are that the word customer in the A/R system is changed to vendor in the A/P system. Of course, the interpretations used when running ACCTPAY are different from those used when running ACCTREC. In the payables program, a payment goes from you to a vendor while in the receivables case the payment goes from the customer to you. In the receivables case, the end of month processing usually is in preparation for sending some statement to the customer. In the payables case this processing helps you,internally.

Because the two programs are so similar we shall illustrate the running procedures only with ACCTREC. When we RUN the program we see, after some initial documentation:

```
          MAIN MENU

1.ADD TO OR BEGIN INVOICE/PAYMENT FILE
2.READ FILE IN FROM TAPE
3.WRITE FILE OUT ONTO TAPE
4.READ FILE IN FROM DISK
5.WRITE FILE OUT ONTO DISK
6.EDIT FILE
7.REPORT GENERATOR
8.POST & ELIMINATE INVOICES-UPDATE FILES
ENTER YOUR CHOICE? <6><ENTER>
```

Before anything is done with this system some valid customers must be put onto the Customer Master File. The file begins with 25(on 16k machines) empty slots for customers(number 1 to 25 on 16k machines). To enter a customer one of these slots must be filled. This is done through the EDIT procedure. We do this and see:

```
EDITING INVOICE FILE(I) OR CUSTOMER FILE(C)? <C><ENTER>

CUSTOMER TO EDIT(0 TO END)?<1><ENTER> (Any number from 1 to 25 would have
     benn OK).

CUST.NO.        CUST.NAME       CURRENT    30-60      60-90       90+
    1           9999              0.00     0.00       0.00       0.00

 PMTS.LAST MONTH     CHGES LAST MONTH     CUMULATIVE CHGES
       0.00                 0.00                0.00

9999                      CUST.NAME     ? <SHENKIN><ENTER>  (Do not make the
     name too long as it be truncated in any reports. 9999 is the number
     given to any customer place not taken up.)

          0.00            CURRENT OWED ?<ENTER> (Just <ENTER> leaves the
     previous data as the new value)
          0.00            30-60 OWED   ? <300.00><ENTER>
          0.00            60-90 OWED   ? <ENTER>
          0.00            90+ OWED     ? <ENTER>
          0.00            PAY LAST MTH ? <300.00><ENTER>
          0.00            CHG LAST MTH ? <ENTER>
          0.00            CUMUL. CHGES ? <750><ENTER>

PRESS ENTER TO CONTINUE ? <ENTER>

CUSTOMER NUMBER TO EDIT(0 TO END)? (We enter the data for several customers.
When we finally decide that we have entered all the data on customers we
choose option 0 on the edit question and return to the MAIN MENU.

After a customer file is generated it is time to input invoices and payments
into the system. This is done with option 1. When we choose this option we
see:

NEW FILE
PRESS ENTER TO CONTINUE?<ENTER>

ENTER NEGATIVE ITEM AMOUNT FOR PAYMENT

                          ITEM NUMBER ? <63><ENTER>
                0         CUSTNUMBER ? <3><ENTER> -(This must be the number
     of an existing customer-otherwise it will be rejected.)
              0.00        ITEM AMOUNT ? <100><ENTER> -(If this amount is
     negative then the program will accept it as a payment, otherwise as a
     charge.
                          COMMENT     ? <FILM><ENTER>
NO.  ITEM #     CUSTOMER #      ITEM AMOUNT     COMMENT
 1   63            3                100.00     FILM
ENTER Y IF OK,N IF ERROR,A TO ABORT?<Y><ENTER>
DO YOU DESIRE MORE ENTRIES(Y/N)? <Y><ENTER>
```

We now add several more invoices. The invoices may be read and changed via the EDIT FILE option on the menu.

Option 7, the REPORT GENERATOR leads to the different reports that the system may handle. When we choose this option we see:

```
                REPORT MENU

1. AGED RECEIVABLES REPORT
2.COMPLETE CUSTOMER LIST
3.COMPLETE CUSTOMER LIST WITH INVOICES ATTACHED
4.LIST OF ALL INVOICES
5.END LIST PROGRAM

ENTER APPROPRIATE CHOICE? <1><ENTER>

          AGED RECEIVABLES REPORT

CUST.NO.  CUST.NAME      CURRENT   30-60     60-90     90+

   1      SHENKIN          0.00    300.00      0.00    0.00
   2      FLIPPER        100.00    125.00    135.00    0.00
   3      GOLDSTEIN        0.00      0.00     65.00    0.00
   4      WOODGLEN       400.00      0.00      0.00    0.00

TOTALS                   500.00    425.00    200.00    0.00

PRESS ENTER TO CONTINUE? <ENTER>
```

The Report Menu now reappears and we may print out any reports we desire. It is highly desirable that the LIST OF ALL INVOICES and the COMPLETE CUSTOMER LIST WITH INVOICES ATTACHED be run before the end of the month as the end of month procedure purges invoices from the system, just posting their totals into the customer records. Thus these reports are the only hard copy available highlighting the month's entire activity. Here is an example of what the list of all invoices looks like:

```
                INVOICE LIST

NO.  ITEM #     CUSTOMER #     ITEM AMOUNT    COMMENT
1    63            3              100.00     FILM
2    50            1               34.98     CHOCOLATE
3    1             1               45.50     BREAD
4    128HJ         4              678.89     CHINA
5    4646          2              -45.50     CK3600  (Taken as payment as
                                                      amount negative)
     TOTAL CHARGES                $859.47
     TOTAL PAYMENTS               -$45.50
     CHG-PAYMENTS                 $813.97

PRESS ENTER TO CONTINUE? <ENTER>
```

When it the end of the month it is time to use option 8 - POST & ELIMINATE INVOICES-UPDATE FILES. When the invoices are entered into the system they do not update the accouint of the customer immediately. Even though they may be printed along with the appropriate customer's record, the customer's totals remain the same. This is the option which does the update. When we choose this option we see:

POSTING ACCTS-PRESS Y TO PROCEED,N TO RETURN? <Y>

ARE OWED BY DATE AND CUMULATIVE BILL TO BE UPDATED(Y/N)? <Y><ENTER>

The payments last month and charges last month columns are always updated by this option. Thus total payments for this month are always available. What Y does here is age all amounts due by 30 days, credit payments to oldest amount due first, add to cumulative charges and then eliminate invoices from the invoice file. So be careful when you choose this option.

An idea is to save the customer and invoice file before doing the end of month procedure. If that stored file is not erased then those invoices are available for examination later.

After the initial setup the files must be read into memory from cassette or diskette **before** being used and must be written back after any changes are made. Both the customer and invoice files are read or written together. Options 2-5 are the options used for storing or recalling data. When using disk there are full options as to what to name your files. Thus a different file could be save for every month. The file saved need not have the same name as the file recalled.

The best way to learn how to use these systems is to practice with them. One word of warning is that to conserve space single precision numeric variables have been used. This means that charaters after the 7th character are not significant. DEFDBL A,B,V in line 58 after CLEAR 2000 will eleviate any problem at the expense of some storage capacity. Only use this option with a machine of more than 16K.

ANNU1

ANNU1 is an annuity computation program. For our purposes an annuity is any set of equal payments made at equal intervals of time. With an ordinary annuity the payments are made at the ends of the periods defined by these intervals while with an annuity due the payments are made at the beginning of these periods. Broadly speaking, the present value of an annuity is the amount of cash which one would accept now instead of the sequence of future payments. Because money has time value (interest is charged) the present value of an annuity does not equal the sum of the payments but is somewhat less, the exact amount depending upon interest rates. The future value of an annuity is the amount which would accumulate by the end of the annuity if the individual payments are invested at the prevailing interest rates. Many annuities also involve a large last payment, called a balloon payment.

In the annuity problems examined by ANNU1 the variables of interest are number of payments, interest rate(either annual or per period), payment per period, present or future value, and balloon payment.

There are four problems addressed:

1. Ordinary annuity- future value
2. Annuity due-future value
3. Ordinary annuity- present value
4. Annuity due- present value

In cases 1 and 2, if we give any three of number of payments,interest rate,payment per period, and future value the program will compute the fourth value. In cases 3 and 4, if we give any 4 of number of payments, interest rate, payment per period, present value, and balloon payment the program will compute the fifth.

Actually the program has alot of interesting uses. In particular, the ordinary annuity-present value option is quite useful in analyzing bank and finance company loans. For example, suppose the proceeds of a loan are $2109.14 to be paid back in 36 installments of 77.00 each. What is the (effective) interest rate being charged. We chose option 3 from the menu and we see:

```
     IS INTEREST RATE QUOTED PER PERIOD(P) OR ANNUALLY(A)? _  -We say A.
     NUMBER OF PAYMENTS PER YEAR? _ - We say 12
 ENTER EACH OF THE FOLLOWING. FOR THE UNKNOWN VALUE ENTER 9999.
     PRESENT VALUE? _  -We enter 2109.14
     PAYMENT? _ - We enter 77.
     NUMBER OF PAYMENTS? _ -We say 36
     ANNUAL INTEREST RATE? _ - We say 9999 as this is the unknown.
     BALLOON PAYMENT? _ - We say 0
```

We then get a summary printout including an interest percentage of 18.7063%. A borrower(or lender) could analyze various loans by changing quantities.

ANNUDEF

An annuity is a sequence of equal payments made at equal intervals of time. Sometimes there is a large balloon payment at the end of the term in addition to the regular last payment. Some examples of annuities are bank loans, mortgage payments,etc. A deferred annuity is an annuity for which the first payment is not made until some fixed time into the future. In order to compare various deferred annuities or to compare a deferred annuity with some other type of financial instrument it is necessary to put them on some type of equal footing. This may be done with present value analysis. Determine the present values of all items for comparison and just compare these present values. **ANNUDEF** will compute the present value of a deferred annuity given the amount of the each individual payment, the number of payments, the interest rate, the time the annuity begins, and the balloon payment, if any.

As an example suppose the interest rate is 10% and we may buy a house with 24 annual payments of $2000 plus a 25th payment of $20000 with the first payment not beginning until year 5. What is the present value of our purchase? This should be the cash equivalent).

We RUN the program and after some initial documentation we see:

```
IS INTEREST RATE QUOTED PER PERIOD(P) OR ANNUALLY(A)? <A><ENTER>
NUMBER OF PAYMENTS PER YEAR? <1><ENTER>
NUMBER OF YEARS INTO FUTURE UNTIL 1ST PAYMENT? <5>

ENTER EACH OF THE FOLLOWING.
ANNUAL UNTEREST RATE(%) ? <10><ENTER>
TOTAL NUMBER OF PAYMENTS ?<25>
ENTER REGULAR PAYMENT? <2000><ENTER>
BALLOON PAYMENT      ? <18000><ENTER> (Note it is in addition to regular)


PRESENT VALUE          = 13534.2
NUMBER OF PAYMENTS     = 25
       CONSISTING OF 1 PAYMENTS ANNUALLY.
ANNUAL INTEREST RATE   = 10
PAYMENT                = 2000
BALLOON PAYMENT        = 18000
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

ARBCOMP

Arbitrage is the buying of an article in one market and the simultaneous selling of it in another market, hopefully at a higher price. In the investment community arbitrage may mean the buying of one security or package of securities and the simultaneous selling of another equivalent package of securities,generally in the same market. This type of arbitrage occurs when mergers between company's are being considered. Generally, in a merger situation one company is really trying to take over another. So the first company offers a combination of stocks, bonds,warrants,cash,etc. to the stockholders of the second company in exchange for a certain number of shares of stock of the second company. The first company generally offers a package of securities worth more than the market value of the second company's stock. Thus the price of the second company's stock tends to go up if people think the merger is going through. The arbitrageur buys stock in the second company(being taken over) and sells an equivalent amount of stock in the first company,shorting it if necessary. If the deal goes through the profit can be tremendous

One of the tasks of the analyst is to figure out precisely what the value of the package of securities offered is. This program will determine this value for any package of securities(up to 100 securities). The necessary input ifor each security in the package is:

The value of 1 unit of that security
The number of units of that security in the package

The package we are talking about could be the securities used to purchase one share of stock in the second company.

We shall now run the program where a package contains .2686 units of common stock in company A and .1930 units of a special Company A bond. A unit of common stock is worth $52.75 while a bond goes for $98.75. What is the value of the package assuming the price of Company A securities will remain essentially constant.

We RUN **ARBCOMP** and see:

```
FOR EACH SECURITY IN PACKAGE ENTER PRICE/UNIT AND NO. UNITS IN PACKAGE.
ENTER VALUE OF 1 UNIT OF SECURITY 1 (-1 TO END)?<52.75><ENTER>
ENTER # OF UNITS OF SECURITY 1 IN PACKAGE?<.2686><ENTER>

ENTER VALUE OF 1 UNIT OF SECURITY 2 (-1 TO END)?<98.75><ENTER>
ENTER # OF UNITS OF SECURITY 2 IN PACKAGE?<.1930><ENTER>

ENTER VALUE OF 1 UNIT OF SECURITY 3 (-1 TO END)? <-1><ENTER>

VALUE OF UNIT       # OF UNITS          TOTAL VALUE
    $52.75            0.2686              $14.169
    $98.75            0.1930              $19.059
==========================================================
           GRAND TOTAL VALUE                $33.227
```

ASSIGN

The assignment problem examined in this program is the following type: There are a fixed number of jobs to be done(say N) and the same number of possible assignees(i.e. those who can do the job. The assignees have different efficiencies for the different jobs. We are given a table giving a cost for each job for each assignee. Each assignee will be assigned to 1 and only 1 job. The idea is to make the assignment which minimizes total cost. This is what **ASSIGN** does. Look at the following job-assignee matrix:

	ASSIGNEE		
JOB	1	2	3
1	25	31	35
2	15	20	24
3	22	19	17

We RUN ASSIGN and after we answer with the number of jobs we enter the following:

```
FOR JOB 1
    COST FOR ASSIGNEE 1 ? <25><ENTER>
    COST FOR ASSIGNEE 2 ? <31><ENTER>
    COST FOR ASSIGNEE 3 ? <35><ENTER>
FOR JOB 2
    COST FOR ASSIGNEE 1 ? <15><ENTER>
    COST FOR ASSIGNEE 2 ? <20><ENTER>
    COST FOR ASSIGNEE 3 ? <24><ENTER>
FOR JOB 3
    COST FOR ASSIGNEE 1 ? <22><ENTER>
    COST FOR ASSIGNEE 2 ? <19><ENTER>
    COST FOR ASSIGNEE 3 ? <17><ENTER>


OPTIMUM ASSIGNMENT
     JOB 1 GETS ASSIGNEE 1   COST = 25
     JOB 2 GETS ASSIGNEE 2   COST = 20
     JOB 3 GETS ASSIGNEE 3   COST = 17

                    TOTAL COST = 62

PRESS ENTER FOR MORE COMPUTATION?
```

AUTOEXP

AUTOEXP will permit the individual to keep a record of all expenses relating to the use of an automobile. A file of expenses is constructed with each entry correspoding to one expense. For each expense the program stores the following:an expense type (chosen from GAS,OIL,MAIN,TOLL,INS,PAYM, or MISC), small description,date,place of service,quantity purchased,bill amount and odometer reading. The expense categories can be used for any class of expense with the exception of GAS. All GAS expenditures are used to compute a miles per gallon rating. The other expense types are summarized. It was the idea that MAIN meant maintenance, INS meant insurance payments, PAYM meant auto loan payments, etc. but the user may use any non-GAS expenses with any of the above categories. When any expense file is begun the program will prompt for an initial odometer reading and date. There must be another entry at the same time that this initial entry is entered. Otherwise, any number of entries may be entered at any time.

When we run the program we see,after some initial documentation:

```
                MAIN MENU

1. ADD TO FILE OR BEGIN NEW FILE
2. READ FILE IN FROM TAPE
3. WRITE FILE OUT ONTO TAPE
4. READ FILE IN FROM DISK
5. WRITE FILE OUT ONTO DISK
6. EDIT FILE
7. LIST CONTENTS, COMPLETE OR BY TYPE
8. ROUTINE ACTUALLY DELETING RECORD FROM FILE

ENTER YOUR CHOICE? <1><ENTER>

EXPENSE TYPES: GAS  OIL  MAIN  TOLL  INS  PAYM  MISC

                       TYPE EXPENSE    ? <INS><ENTER>
                       EXP. DESCRIPTION? <JAN PAY><ENTER>
                       DATE OF EXPENSE ? <10/03/80><ENTER>
                       PLACE OF SERVICE? <ENTER>
      0.00             QUANT. PURCH    ? <ENTER>
      0.00             BILL AMOUNT     ? <77.39><ENTER>
      0.00             ODOMETER READ.  ? <ENTER>
```

The entry is then displayed on the screen on a single line with athe user given the choice of accepting or rejecting the entry. What is seen is:

```
TYPE  DESCRIPT  DATE      PLACE   QUANT   $ AMT.    ODOMETER
INS   JAN PAY   10/03/80            0.0    77.39        0.0
ENTER Y IF OK,N IF ERROR,A TO ABORT
```

A Y puts the record in the file, an N allows the user to correct parts of the entry, while A eliminates the entry & returns to the MAIN MENU

Of the entries which were made only the TYPE and $ AMT. were important as these are used to help summarize in the report. For GAS the quantity is important as miles per gallon is computed. By the way, if liters are used for amount of GAS then miles per liter will be computed even if the printout shows miles per gallon. DO NOT mix liters and gallons for quantity as this will mess up the computations.

As an example of a complete list of the expense file, gotten through option 7 we have the following:

COMPLETE LIST

TYPE	DESCRIPTION	DATE	PLACE	QUANTITY	$ AMT.	ODOMETER
GAS	INITIAL	09/24/80		0.0	$0.00	24567.7
GAS	BUY GAS	09/27/80	NJ	15.6	$18.00	24765.2
TOLL	BRIDGE	09/27/80	GW	1.0	$1.50	0.0
TOLL	BRIDGE	09/28/80	GW	1.0	$1.50	0.0
GAS	BUY GAS	09/29/80	NANUET	16.0	$18.00	25030.0
INS	JAN PAY	09/29/80		0.0	$77.39	0.0

TOTAL GASOLINE USED	:	31.60
TOTAL MILEAGE GONE	:	462.30
TOTAL COST OF USE	:	$116.39
MILES/GALLON	:	14.63
TOTAL COST/MILE	:	$0.25

The lists may be printed on either CRT or line printer, although sometimes, due to the limited size of the CRT the reports are somewhat compressed.

When using the EDIT FILE option each record is listed along with an item number to its left. This item number permits the user to access the record for the edit when the program prints:

ENTER NUMBER OF ITEM YOU WISH TO EDIT,0 IF NONE?

The edit itself is just like the data entry with the exception that the user may see what the previous value of the entry was in the edit mode. When editing, an expense type of ZZZZ will mark an item for deletion but will not actually do the deletion. That is done by the last option on the MAIN MENU.

All files must be saved on diskette or cassette after use or modification if future manipulations are to be required. With disk files the user may name the expense file so several files may be handled on one diskette.

On a 16K machine the present DIM statements should provide entries for 70 expenses. On machines with more memory many more expenses may be handled. The user need only change the DIM statements near the program's beginning.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BAYES

The ideas of conditional probability are used in much of decision making theory. A common situation is the following: There are several possible real-world 'states of nature'. One of these is the 'true' state. For example, an oil well may yield no oil, from 1-100 barrels per day, from 101-1000 barrels per day or more than 1000 barrels per day. Before we drill the well we don't known which is the 'true' state. What we may know is the probability of occurrence of each state of nature. For example, these probabilities may come from geological considerations in the oil well example. There is also some activity we wish to perform, e.g. drill 1000 feet of the oil well and look at more samples. We assume that we know the conditional probability of a special result of the activity, given each of the states of nature. In the oil-well example this result might be that a certain type of mud is found in the well. Now suppose the result does in fact occur. What the present program does is use a result called BAYES' ThEOREM to give us after the fact revised estimates of the probabilties of the possible states of nature. These revised probabilties form what is known as a posterior distribution and are used in decision making quite often.

The following example shows how the program runs: There are 3 states of nature. These are market up,market down, and market unchanged. Let B be the event that I make money. I know that if the market is up I will make money with probability .8, if down I will make money with prob .1 and if unchanged I will make money with probability .4 . There is a 30,30,40 chance of the market going up,remaining the same, and going down respectively. If it winds up that I made money then what are the probabilties that the market was up,even, and down respectively? We run the program and see:

```
ENTER PROB. ST. OF NATURE 1(-1 TO END)?<30><ENTER>
ENTER PROB. ST. OF NATURE 2(-1 TO END)?<30><ENTER>
ENTER PROB. ST. OF NATURE 3(-1 TO END)?<40><ENTER>
ENTER PROB. ST. OF NATURE 4(-1 TO END)?<-1><ENTER>

PROBABILTIIES SUM TO 100. THIS SHOULD BE STANDARDIZED
TO SUM TO 1. ENTER 1 TO STANDARDIZE,0 TO END? <1><ENTER>  (Each prob/100)

ENTER COND.PROB. OF EVENT B GIVEN ST.OF.NATURE 1 ?<.8><ENTER>
ENTER COND.PROB. OF EVENT B GIVEN ST.OF.NATURE 2 ?<.4><ENTER>
ENTER COND.PROB. OF EVENT B GIVEN ST.OF.NATURE 3 ?<.1><ENTER>
```

S OF N	ORIG PROB	PROB B GIVEN S OF N	POSTERIOR PROB S OF N GIVEN B
1	0.300	0.800	0.600
2	0.300	0.400	0.300
3	0.400	0.100	0.100

The posterior probabilities may be used with expected values to compute the revised expected profit of a project. These are after the fact probabilties.

BETAALPH

Given data with regards to prices and dividends over a period of time for a given stock and an investor selected market index (e.g. S&P 500) this program will estimate the alpha and beta values for this stock during the given time period. Necessary input begins with the number of years(or periods) for which data is available. For each of these periods we input the market index price plus any dividends and the final price for the chosen stock along with any dividends paid.

Output includes alpha, beta, an expected rate of return for the security, the variance of return and the standard deviation of return. Some data of interest to statisticians is also generated. We run an example for a security during 6 periods with the following information:

Period	security		market	
	price	dividend	price	dividend
1	28	1	101	5.45
2	30	1	104	6.28
3	31	2	105	6.15
4	35.125	2.50	110	7.00
5	35	2.50	112	6.98
6	38	2.75	115	7.11

We RUN the program and after some initial documentation we see:

```
HOW MANY PERIODS OF DATA ARE AVAILABLE ? <6><ENTER>

INPUT MARKET INDEX PRICE,DIVIDEND FOR PERIOD 1 ? <101,5.45><ENTER>
----------
INPUT MARKET INDEX PRICE,DIVIDEND FOR PERIOD 6 ? <115,7.11><ENTER>

INPUT STOCK PRICE,DIVIDEND FOR PERIOD 1 ? <28,1><ENTER>
-----
INPUT STOCK PRICE,DIVIDEND FOR PERIOD 6 ? <38,2.75><ENTER>
```

At this time the values are computed and we see:

```
THE NUMBER OF OBSERVATIONS EQUALS 6
THE ESTIMATED EQUATION IS Y =  2.840 X   +   -0.123
THE CORRELATION COEFFICIENT =0.8154
-------
ALPHA           = -.123208
BETA            = 2.83961
EXPECTED RETURN=  .130551
```

The last line says we may expect a 13% average return per period.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BLACKSH

An option is the right to buy(call option) or sell(put option) a specific security at a specified price during some predetermined time period. Options may be bought and soldl The price at which an option sells depends upon many factors, the most important of which are current price of the underlying security, the exercise price of the option (the price at which the holder of the option may buy or sell the security), the price volatility of the underlying security, and the prevailing interest rate. Based upon the above variables **BLACKSH** will compute the value of a put or call option. The computation procedure used involves the well known Black-Scholes option evaluation formula.

Input includes stock price, option exercise price, interest rate, days to expiration of the option, and stock price volatility. There is some debate as to which interest rate should be used. Some use the broker loan rate, others use various traesury securities rates. Volatility may be determined through examination of various broker and advisory service literature.

Output is the current value of the option plus a hedge ratio. The hedge ratio is the number of shares of stock which can be sold with the purchase of one call option to leave the purchaser with a ´risk free´ situation.

We will show how the program operates by using the following problem: A share of GHJ stock is now selling for $45. The volatility of GHJ shaares is 30%. The current interest rate is 11%. How much is a call option with an exercise price of 50 and with 120 days left until expiration.

We RUN the program and after some intial documentation we see:

```
STOCK PRICE,EXERCISE PR.,INT RATE(%),DAYS TO EXP,VOLATILITY(%)
? <45,50,11,120,30><ENTER>

ARE OPTIONS PUTS(P) OR CALLS(C)? <C><ENTER>

ST.PR.   EX.PR.    INT.RT.    D.T.EXP.    VOL.     VALUE   HEDGE

45.00    50.00     0.11        120       0.30      1.84    0.38

PRESS ENTER FOR MORE?
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BONDVAL

When considering the value of bonds, items which should be considered include yield to maturity, time to maturity, maturity value, current price, and coupon interest rate. Given any four of these this program will compute the fifth. In addition, if desired, income taxes, capital gains taxes and commissions may be considered.

As an example look at the following: ACE Corporation 7% bonds mature in five years after 10 more interest payments. Joe can buy ACE bonds at a price of 93 (i.e. $930). What will the annual yield to maturity be excluding taxes and commission considerations?

We RUN **BONDVAL** and after some initial documentation we see:

```
INPUT THE FOLLOWING AMOUNTS. PRESS ENTER WHEN AN
ITEM WILL BE COMPUTED. INPUT 0 IF AN AN ITEM IS
NOT TO BE CONSIDERED. POSSIBLE FIELDS FOR
NONCONSIDERATION ARE COMMISSION, CAPITAL GAINS
AND INCOME TAX RATES.

PURCHASE PRICE PER BOND                          ? <930><ENTER>
COMMISSION PER BOND                              ? <0><ENTER>
MATURITY VALUE                                   ? <1000><ENTER>
NUMBER OF INTEREST PAYMENTS PER YEAR             ? <2><ENTER>
NUMBER OF PAYMENT PERIODS TO MATURITY            ? <10><ENTER>
ANNUAL COUPON INTEREST RATE(%)                   ? <7><ENTER>
ANNUAL YIELD TO MATURITY                         ?<ENTER>  (want to find)
CAPITAL GAINS TAX RATE                           ?<0><ENTER>
INCOME TAX RATE                                  ?<0><ENTER>

ENTER THE NUMBER OF THE VARIABLE TO BE COMPUTED
1. CURRENT PRICE PLUS COMMISSION
2. MATURITY VALUE OF THE BOND
3. COUPON INTEREST RATE
4. YIELD TO MATURITY OF THE BOND
5. LENGTH OF TIME TO MATURITY
MAKE CHOICE ? <4><ENTER>

1.CURRENT PRICE PLUS COMMISSION                  : 0
2.MATURITY VALUE OF THE BOND                     : 1000
3.ANNUAL COUPON INTEREST RATE                    : 7
4.YIELD TO MATURITY OF BOND                      : 8.75637
5.LENGTH OF TIME TO MATURITY(PERIODS)            : 10
6.INCOME TAX RATE                                : 0
7.CAPITAL GAINS TAX RATE                         : 0
8.NUMBER OF INTEREST PAYMENTS PER YEAR           : 2

PRESS ENTER FOR MORE COMPUTATION ?
```

BONDVAL2

Many people consider a bond investment to be a conservative type investment. This is true to some extent since if the company is sound and if the bond is held to maturity then the owner will be paid the maturity value of the bond at maturity plus individual interest payments all through the life of the bond. However, if a bond must be cashed in before maturity trouble may ensue. If interest rates are very high and the bond is relatively long term then the price of that bond would tend to be quite low. Likewise if interest rates are very low then the value of the long term bond tends to be quite high.

In this program the effect of interest rates on bond prices is examined through construction of a table listing 10 interest rates along with the present value of the corresponding bond payments. We show how the program operates with the analysis of a bond with 12 years to maturity, earning $100 per bond per year for interest, and with a maturity value of $1000

We RUN the program and see:

```
NUMBER OF YEARS TO MATURITY            ?<12><ENTER>
COUPON INTEREST PAYMENT PER YEAR       ?<100><ENTER>
MATURITY VALUE OF THE BOND             ?<1000><ENTER>
LOWEST INTEREST RATE TO CONSIDER(%)    ?<7><ENTER>
HIGHEST INTEREST RATE TO CONSIDER(%)   ?<16><ENTER>

DO YOU DESIRE LINE PRINTER OUTPUT(Y/N);:<Y><ENTER>

 12 YEARS TO MATURITY
COUPON PAYS $100 PER YEARC FOR INTEREST.
MATURITY VALUE FOR THE BOND IS 1000
INTEREST RATE      VALUE OF BOND
    7.0              1238.28
    7.9              1159.08
    8.8              1086.80
    9.7              1020.74
   10.6               960.29
   11.5               904.89
   12.4               854.05
   13.3               807.33
   14.2               764.34
   15.1               724.72

PRESS ENTER FOR MORE COMPUTATION ?
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

BREAKEVN

As anyone who has run a business knows there are certain costs which do not change no matter how good or bad business is. Some examples of these are rent, payments on old bank loans, salaries of key nonproduction employees who cannot be terminated, cost of some heat, etc. These types of costs are called fixed costs. When items start being produced the businessman discovers that he has certain other costs. These include the charges for the materials which went into these items, the salaries of the production employees making the item, extra power costs needed to make the item, etc. These costs are variable costs. If a company makes no items during a year then it is still liable for its fixed costs. As items are made, if the variable cost of that item is less than the cost of the item, the company begins to make up for the fixed costs. At some point the company is just even. If it produces even more of the item then it begins to make money. What the current program will compute is the volume of sales needed for the company to reach the break even point, i.e. the point beyond which the company will be making a profit on the item.

Lets see how the program works with a simple example:

We have fixed costs of $50000. The item being manufactured sells for $12.50 each but has variable parts and labor costs of $8.00. How many of these items must I sell to break even?

We run the program and after some initial documentation we see:

```
INPUT TOTAL FIXED COSTS IN DOLLARS ?_ We press 50000 <ENTER>

INPUT UNIT SELLING PRICE IN DOLLARS ?_ We press 12.50 <ENTER>

INPUT VARIABLE COST PER UNIT IN DOLLARS ?_ We press 8.00 <ENTER>


BREAK EVEN POINT IS 11111.1 UNITS

TO LOOK FOR GIVEN PROFIT POINT INPUT 1
TO DO ANOTHER BREAK EVEN ANALYSIS INPUT 2
?_ We press 1 <ENTER>

WHAT PROFIT DO YOU DESIRE ?_  We press 100000 <ENTER>
SALES NEEDED FOR PROFIT 100000 EQUALS 33333.3 UNITS
INPUT 1 FOR ANOTHER BREAKEVEN ANALYSIS?_
```

Note that our answers may mathematically come out as decimals even when it is impossible to have 1/2 a person or .1 item used, etc. Don't worry but just round the answers.

BUSBUD

This simple bookkeeping program has the same categories as those of the DOME SIMPLIFIED WEEKLY BOOKKEEPING RECORD No. 600 available nationwide in stationary and book stores. (This is the brown DOME book. (copyright by DOME ENTERPRISES) The program has the ability to handle one month's worth of data at a time. This is different from the DOME system which is weekly oriented. During the month the data may be stored on cassette or disk. On the disk the different months' data will be stored automatically in different files with the names MONTH01, MONTH02,etc. On cassette data for different months should be kept on different cassettes.

When first using the system the businessman should choose option 7 of the main MENU. This allows one to review the various categories which into which income and expenses may be divided. By the way (see the DOME book) category 0 represents income while all higher numbered categories represent expenses. This system differs from the DOME system in several important respects. The DOME system keeps year to date totals while this system does not in the present version. However, this system gives the user the ability to enter a budgeted amount for each category and to compare results against budget. The DOME system provides for payroll while this system does not. However, the PAYROLL2 program will handle a small business payroll. After looking at the various categories it is time to enter a budgeted amount for each category. Still in the option 7 submenu just choose the CHANGE BUDGET AMOUNT FOR MONTH option and enter a budget amount for each category which appears. If you are not used to budgeting justy 0 could be entered for each category budget amount. On returning to the MAIN MENU be sure to WRITE THE MONTH's DATA to disk or cassette. Nothing is automatically saved or read into memory. These operations must be explicitly done using MAIN MENU options 1 to 4.

After initially setting up the budget amounts the user should run the system in the following manner:

1. Load BUSBUD and RUN it.
2. READ the month's DATA in from diskette or cassette
3. ADD to the month's file.
4. Make any EDIT changes to your previous entries.
5. Do a BUDGET ANALYSIS (option 8)
6. SAVE the month's data onto diskette or cassette.

NEVER forget to save the data. At best you will lose the latest entries while at worst you will lose your entire month's file.

Each entry to a month's file takes the form:

CLASSIFICATION,AMOUNT,DAY? _

The CLASSIFICATION is the category number, the DAY is the day of the month for the income or expense. Dimensioning on the distributed version of the program allows 100 entries per month but this may be done as per the documentation in the program itself.

To learn this system, use it. Printed analyses are not supported on the disributed program but should prove very simple to add.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

CAP1

CAP1 uses the Capital Asset Pricing Model to compute the return an investment project should yield. The Capital Asset Pricing Model studies the risk-return impact of an investment proposal upon an investor's portfolio of stocks and other investments. The model yields the expected return for an investment given the expected yield for the maket as a whole, the (risk-free) interest rate for U.S. Treasury securities, and the volatility of the return on the investment with respect to the rest of the market. Volatility is also known as beta. A volatility of 1 means that the investment always yields just about the same amount as the market as a whole. A volatility greater than 1 means that the yield on the investment moves in the same direction as that of the market but in faster and greater swings. A volatility of 0 means that the return on the investment is independent of what happens in the market as a whole. A negative volatility means that return on the investment and that of the market move in opposite directions, i.e. when one goes up the other goes down. For publicly traded securities the volatility may be ascertained by examining the broker literature. For other investment proposals this method is not so useful and this information must be somehow computed.

To see how the program works we shall examine the following problem:

> Joe is considering an investment in an old brownstone. He guesses that this investment has a volatility of 1.32. The market is expected to yield 11.2 % during the time Joe would own the brownstone. Treasury Bills are expected to yield about 8.3% during this time. What return should Joe expect from his investment? What if the market yields 16%?

We run the program and after some initial documentation we see:

```
ENTER TREASURY INTEREST RATE(%) ? <8.3><ENTER>
ENTER EXPECTED MARKET RETURN(%) ? <11.2><ENTER>
ENTER VOLATILITY FOR INVESTMENT ? <1.32><ENTER>
THE EXPECTED RETURN ON THIS INVESTMENT IS 12.128%

PRESS ENTER FOR MORE COMPUTATIONS ? <ENTER>

ENTER TREASURY INTEREST RATE(%) ? <ENTER>  (Just <ENTER> retains last value)
ENTER EXPECTED MARKET RETURN(%) ? <16><ENTER>
ENTER VOLATILITY FOR INVESTMENT ? <ENTER>
THE EXPECTED RETURN ON THIS INVESTMENT IS 18.464%

PRESS ENTER FOR MORE COMPUTATIONS ?
```

CHECK2

This program will permit the user to input the necessary data and to print checks on a line printer, if available. Checks and stubs put out by New England Business Service, Inc. were the test checks. In particular, we used their Product 9020. In the following, we shall show how to use the program by actually running an example. For illustration only, user input shall be enclosed in < > symbols. When we run the program we see:

```
                MENU

1. INPUT DATA AND PRINT CHECKS
2. PRINT CHECK REGISTER
3. INPUT TODAY'S DATE INTO SYSTEM

ENTER CHOICE ?_ <3><ENTER>    (Note that this is the default check date and
                    will be the date on the check stub.

ENTER TODAY'S DATE IN FORM MM/DD/YY?_ <07/30/80><ENTER>

SYSTEM DATE IS 07/30/80
PRESS ENTER TO CONTINUE ?_ <ENTER>  (We go back to the MENU. This time we
                    use option 1 to input some checks.

CHECK DATE(0 TO END) ?_ <08/01/80><ENTER>  (Just enter here for system date)
INPUT CHECK # ?_ <1050><ENTER>
INPUT AMT OF CHECK ?_ <150.09><ENTER>
ENTER PAYABLE TO ?_ <H & E COMPUTRONICS><ENTER> (At most 22 characters)
ENTER COMMENTS(LINE 1)?_ <FOR NEWDOS/80 $149.00><ENTER>
ENTER COMMENTS(LINE 2)?_ <SPECIAL CHARGE $1.09><ENTER>

CHECK DATE 08/01/80
CHECK #    1050
CHECK AMOUNT 150.09
PAYABLE TO H & E COMPUTRONICS INC
COMMENT 1   NEWDOS 80   $149.00
COMMENT 2   SPECIAL CHARGE  $1.09

ENTER Y IF OK, N IF ERROR <Y><ENTER> (If we enter N the program returns us
                        to the input routine.)
ENTER Y TO PRINT CHECK, N TO SKIP PRINT ?_ <Y><ENTER> (The check is now
                        printed and the program returns for more input.)

CHECK DATE(0 TO END)?_ <0><ENTER> (We return to the menu and print a
                        register, option 2.)

CHECK #        DATE          PAYABLE TO                AMOUNT
1050         08/01/80  H & E COMPUTRONICS INC          150.09

TOTAL CHECKS = 1     TOTAL CHECK AMOUNTS =   $150.09
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

CHECKBK1

This program acts as a check register. All checks, deposits, and miscellaneous charges may be entered and stored. The "checkbook" may be stored on either cassette or diskette. It is even possible to transfer the checkbook from cassette to diskette when a diskette drive is purchased.

What can the system do? For each entry the following is stored:

1. A number. In the case of checks this automatically sequences from a given beginning check#. For deposits this number = -1 and for miscellaneous charges it = -2
2. A description. For checks this is usually the payee,for deposits what was deposited.
3. A category. This may be a number from 00 to 99. There is an option in the program to list and total all items in any particular category.
4. An amount of check, charge or deposit
5. The current balance, i.e. the balance after the current transaction.

If a line printer is attached a listing of the checkbook may be provided. This listing is important as it also lists the item # of each item in the checkbook. This item # is the order in which the item was entered into the checkbook. It is used in the edit option where the data on any item may be changed. If any dollar amount is changed in the edit then the current balance for all other items is adjusted. It is possible to edit without a line printed report by "searching" for the appropriate item #. This should never take more than a few seconds if you first print out your checkbook on the screen.

The only real disadvantage of this checkbook system is that it is an in memory system, i.e. the checkbook must be loaded into memory from storage before any manipulations may be done. When the program is first run the option on the MENU for setting up a new checkbook is run. This allows the user to enter the initial balance, initial date, and the initial check # of the new checkbook. When all entries are completed for this initial run and all appropriate printouts examined then it is time to save the checkbook on cassette or diskette(option 4 or 9 from MENU). In all subsequent runs the checkbook must first be loaded and then new entries added, edits made, printouts done, etc.

Being an in memory system there is a limitation as to the number of items in a checkbook atr one time. On a machine with 16K of free RAM memory this should be about 120 items. 32K would give about 300 items. This is not too limiting as a new checkbook may be started every month or quarter. A warning is that the number variables are single precision and so dollars may not add up if alot is kept in the account(e.g. $10,000 + on TRS-80).

COMPBAL

When a businessman borrows from a bank one of the requirements may be the maintenance of a compensating balance. A compensating balance is a certain proportion of the loan which cannot be taken out of the businessman's checking account. For example if the businessman borrows $1,000,000 with a 20% compensating balance requirement then he may only write checks on $800,000 of the loan. The remaining $200,000 is the compensating balance. Since the businessman is paying interest on the entire $1,000,000 the effect of this is that he pays more than the nominal interest rate for the loan. For example if the above loan was at a 10% nominal rate the the businessman would be paying $100,000 interest to have use of $800,000 of a true 12.5%. Actually, things aren't quite this bad. Usually a business keeps a certain minimum in its checking acount anyway. This amount can be part of the compensating balance. Thus, in the above, if the business had at least $60,000 on deposit at all times anyway only $140,000 of the loan would be needed for the compensating balance. This raises the interest rate above the worst case situation of above.

COMPBAL does the above computations for loans involving compensating balances. Input consists of proceeds of the loan, compensating balance percentage, nominal interest rate on the loan, and the amount normally kept as a minimum in the checking account. Output consists of the nominal compensating balance amount(corresponding to the $200,000 above), true compensating balance amount(corresponding to the $140,000 above), the available funds from the loan(proceeds - true compensating balance), and the effective interest rate for the loan.

In the following example a business borrows $130,000 with a 10% compensating balance at 13%. Normally that business keeps a minimum of $10,000 in its checking account. What is the effective interest rate of the loan?

We run **COMPBAL** and after some initial documentation we see:

```
INPUT PROCEEDS OF LOAN IN $              ? <130000><ENTER>
INPUT COMPENSATING BALANCE %             ? <10><ENTER>
INPUT INTEREST RATE ON LOAN (%)          ? <13><ENTER>
NORMAL ACCOUNT BALANCE WITH NO LOAN      ? <10000><ENTER>

NOMINAL COMPENSATING BALANCE AMOUNT          $130000.00
TRUE COMPENSATING BALANCE AMOUNT               $3000.00
AVAILABLE FUNDS FROM LOAN                    $127000.00
EFFECTIVE INTEREST RATE                           13.31 %

PRESS ENTER FOR MORE COMPUTATION? _
```

CONDPROF

This program assumes that there are several possible states of nature. There are also several actions which may be taken. If the state of nature(S of n) is known then each action leads to a known monetary result. This result is called a conditional profit. For example,if the payoff when action 3 is taken and S of N 2 is true equals $35 then $35 is the conditional profit of action 3 given S of N 2. Each action-S of N pair yields a conditional profit. This program constructs a so-called conditional profit table, which is just a rectangular matrix, the elements of which are conditional profits. We also assume that we know the probability of occurrence of each S of N and compute the expected payoff for each action.

In the following a conditional profit table is constructed in a case with two possible S of N's and 3 possible actions. S of N 1 occurs with probability .3 and S of N 2 with probability .7. When S of N 1 is true then our payoffs are 2 for action 1, 5 for action 2, and 3 for action 3. When S of N 2 is true the payoffs are 5 for action 1,7 for action 2 and 2 for action 3. We RUN the program and see:

```
ENTER NUMBER OF POSSIBLE STATES OF NATURE? <2><ENTER>
ENTER PROBABILITY OF S OF N 1 ? <.3><ENTER>
ENTER PROBABILITY OF S OF N 2 ? <.7><ENTER>

ENTER NUMBER OF POSSIBLE ACTIONS(<=5)? <3><ENTER>
NO PAYOFF MORE THAN 9999.99

ACTION 1
     PAYOFF WHEN S OF N IS 1 ? <2><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <5><ENTER>
ACTION 2
     PAYOFF WHEN S OF N IS 1 ? <5><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <7><ENTER>
ACTION 3
     PAYOFF WHEN S OF N IS 1 ? <3><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <2><ENTER>

     CONDITIONAL PAYOFFS OF ACTION GIVEN S OF N

S N  PR                      ACTION
          ACTION 1      ACTION 2      ACTION 3
 1  0.30      2.00          5.00          3.00
 2  0.70      5.00          7.00          2.00

EXP VAL       4.10          6.40          2.30
```

CVP

A businessman's profits depends on his costs and his volume of business. Two common questions every businessman asks are: How much will I earn if my I know exactly what my sales will be? and What must my sales be for me to earn a certain amount I will specify. **CVP** does the computations which permits both of these questions to be answered.

Necessary input includes total fixed costs for the enterprise, income tax rate on profits as a per cent, the ratio of variable cost per unit to sales price per unit, and either specified sales or specified income. The computer computes the other variable.

Lets try this with the following example:

Kerri's Place has fixed costs of $4500 per month. Kerri sells honey cakes for $5 each with unit variable costs of $3. The tax rate for Kerri is 48%. How much must Kerri sell to earn $30,000 per month? What will Kerri earn if she sells $100,000 worth of honey cakes per month?

We run CVP and see after some initial documentation:

```
      MENU

1. DETERMINE SALES FOR SPECIFIED INCOME
2. DETERMINE INCOM FOR SPECIFIED SALES

ENTER YOUR CHOICE ? <1><ENTER>

ENTER TOTAL FIXED COSTS IN DOLLARS    ? <4500><ENTER>
ENTER TAX RATE AS PER CENT            ? <48><ENTER>
ENTER RATIO OF VARIABLE COSTS PER UNIT
 TO SALES PRICE PER UNIT              ? <.60><ENTER>
ENTER DESIRED INCOME LEVEL            ? <30000><ENTER>
TO REACH INCOME LEVEL $30000.00 YOU MUST SELL $155481.00
IN GOODS

PRESS ENTER TO CONTINUE ? <ENTER>
```

We return to the Menu. Choice 2 is now chosen. The questions are the the same but instead if ENTER DESIRED INCOME LEVEL we see:

```
ENTER SPECIFIED SALES LEVEL          ? <100000>
FOR SALES LEVEL $100000.00 YOUR INCOME LEVEL WILL EQUAL $18460.00

PRESS ENTER TO CONTINUE?
```

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

DATE

This short program computes the number of days between any two given dates as long as both dates involve time after the year 1582. Why would anyone want to know this type of information? One possible application is to discover how old you are. For example suppose Karen was born on January 13, 1958 and it is now July 12, 1980. How many days old is Karen? We run this program and determine that Karen is 8216 days old. Children seem to love this type of result. The real application of this computation is in the business world where it is necessary to do things like determining the number of days left to the maturity of a bond, calculating how many days money was left in a bank account with day of deposit to day of withdrawal accounts, etc.

Using the program is very simple. We just wish to determine the number of days from a date called DATE 1 to a date called DATE 2. We RUN the program and after some initial documentation we see:

INPUT DATE 1 IN FORM MM,DD,YYYY (We enter 01,13,1958 - note the commas)
We press <ENTER>

INPUT DATE 2 IN FORM MM,DD,YYYY (We enter 07,12,1980 and press <ENTER>)

We see:

THE NUMBER OF DAYS BETWEEN 1/13/1958 AND
7/12/1980 EQUALS 8216

PRESS 1 FOR MORE TRIALS, 0 TO END ? _ (We press 0 and <ENTER> to end.

Of particular interest to programmers is the fact that this program is written so that it may easily be converted into a subroutine for use as part of a longer and more complex program, e.g. a day of deposit to day of withdrawal bank routine. The instructions as to which lines to change and delete are in the actual program code as REM statements. When used as a subroutine the use must do the following:

1. The month,day and year of date 1 must be input into the subroutine from elsewhere in the program as the variables M(1), D(1), and Y(1) respectively.

2. The month,day and year of date 2 must be input into the subroutine from elsewhere in the program as the variables M(2), D(2), and Y(2) respectively.

3. The variable F(3), returned from the subroutine is the desired number of days between date 1 and date 2.

DATEVAL

A dated value is a pair consisting of the maturity value of an obligation along with its date of maturity. For example, $750 due on September 13, 1980 is an example of a dated value. Suppose we wanted to compare the above dated value with $900 due on September 13, 1982. It does not really make sense to say the the second dated value is $150 more than the first since there is a two year time difference. In order to determine which of these dated values we prefer it will be necessary to pick an interest rate and discount one of these dated values to the time period of the other. When two dated values have the same maturity date then we can them simply by using their maturity values.

DATEVAL allows us to compare dated values. We are given a dated value maturing some fixed number of years into the future. An interest rate to be used in the comparison is also supplied. Then if we input some other fixed time in the future the program will find an amount so the amount and the input date will give a dated value equivalent to the original dated value. Two dated value are equivalent if when discounted to the same time period their amounts are the same.

The following example shows how the program operates: A debt of $1000 is due at the end of 10 years. If the interest rate is 11% then find equivalent debts at the end of (a) 2 years (b) 15 years.

We RUN the program and after some initial documentation we see:

```
ENTER INTEREST RATE(%) TO BE USED IN COMPARISON? <11><ENTER>

INPUT FOR GIVEN DATED VALUE
          INPUT MATURITY VALUE ? <1000><ENTER>
          INPUT # OF YEARS INTO THE FUTURE FOR MATURITY? <10><ENTER>

INPUT # OF YEARS INTO FUTURE FOR MATURITY OF DATED VALUE FOR
COMPARISON? <2><ENTER>

1000 DUE IN 10 YEARS IS EQUIVALENT TO 433.926 DUE IN 2 YEARS

PRESS 1 TO KEEP SAME ORIGINAL DATED VALUE, 2 TO BEGIN OVER?<1><ENTER>

INPUT # OF YEARS INTO FUTURE FOR MATURITY OF DATED VALUE FOR
COMPARISON? <15><ENTER>

1000  DUE IN 10 YEARS IS EQUIVALENT TO 1685.06 DUE IN 15 YEARS.

PRESS 1 TO KEEP SAME ORIGINAL DATED VALUE, 2 TO BEGIN OVER?
```

DAYYEAR

This is a very simple program which computes the day of the year on which any particular date falls. Input consists of the number of the month within which the desired date falls and the day of the month it is. Output consists of the day of the year the date falls upon in normal years and in leap years. The program will also determine how many days separate two dates during the same year.

We wish to determine the day of the year on which February 15 occurs, the day of the year on which July 17 occurs and the number of days from February 15 to March 3.

We RUN the program and after some initial documentation we see:

```
ENTER MONTH OF DESIRED DATE IN NUMERICAL MM FORM? <2><ENTER>
ENTER DAY OF FEBRUARY DESIRED? <15><ENTER>

FEBRUARY 15 IS DAY NUMBER 46 OF THE YEAR

NUMBER OF DAYS FROM LAST DATE = 46

PRESS ENTER FOR MORE COMPUTATIONS? <ENTER>


ENTER MONTH OF DESIRED DATE IN NUMERICAL MM FORM? <7><ENTER>
ENTER DAY OF JULY DESIRED? <17><ENTER>

JULY 17 IS DAY NUMBER 198 OF THE YEAR.

DURING A LEAP YEAR JULY 17 IS DAY NUMBER 199 OF THE YEAR.

NUMBER OF DAYS FROM LAST DATE =  152
NUMBER OF DAYS FROM LAST DATE DURING LEAP YEAR = 153

PRESS ENTER FOR MORE COMPUTATIONS?
```

There are, during a normal year, 152 days from February 15 to July 17. If the later date comes first then the last two numbers printed are negative.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

DEPLETE

Depletion refers to exhausting natural reseouces, e.g. the oil in a well or the copper in a mine. In usual methods of depreciation the exact depreciation schedule is determined soon after the item is purchased. This is done by estimating a productive lifetime and a salvage value at the end of the productive lifetime. Then a formula is used. There is a different proceedure used by companies in the nonrenewable resource mining business. Here the property in question is a mine, a well, etc. This proceedure for computing depreciation will be called the depletion method and works as follows:

The owner estimates what the total productive output of the property will be over the useful life of the property. He also takes his initial cost for the project, estimated salvage value, estimated total repair cost over lifetime of the project. With this a number is computed, say A. Then, the depreciation allowance in a year with production level x will be Ax or A times x. This shows that the depletion method of computing depreciation makes the depreciation amount proportional to volume of sales for the year.

Given the above data **DEPLETE** will compute different depreciation amounts depending directly upon volume of production for that year.

We now run **deplete** with the following example:

Ewingg Oil just bought a new well from Saudi Arabia. The well and land costs $2,500,000, have a final salvage value of $500,000. It is estimated that total repair cost over the project's lifetime will be $300,000. It seems that the well will produce a total of 2,000,000 barrels over its lifetime . What is depreciation expense for the year in which production was $400,000

We RUN the program and see:

```
COST OF PURCHASE AND DEVELOPMENT ? <2500000><ENTER>
FINAL SALVAGE VALUE ESTIMATE     ? <500000><ENTER>
TOTAL REPAIR COST OVER LIFE      ? <300000><ENTER>
ESTIMATE OF LIFETIME PRODUCTIVE OUTPUT?<2000000><ENTER>

ENTER PRODUCTION LEVEL(-1 TO END)1000
DEPRECIATION CHARGES ARE $1150.00

ENTER 1 FOR NEW PRODUCTION LEVEL,2 FOR NEW ITEM, 3 TO END?
```

DEPRDB

One of the expenses of operating a business is the loss in value of physical property due to factors such as wear and tear, deterioration and decay, obsolescence, etc. Such a loss in value is called depreciation and is generally treated as an expense on a company's income statement. All other things being equal, greater depreciation charged against an asset in a given year will mean lower profits for the corporation. However, lower profits mean lower taxes so the higher depreciation may actually mean that the company is better off cash wise even though it has lower profits. PP CORP. earned $100000 before taxes and depreciation charges and is in the 50% tax bracket. If depreciation is $10000 then the company earns $90000, pays $45000 in taxes and keeps $45000 + $10000 = $55000. If depreciation is $20000 then the company earns $80000, pays $40000 in taxes and keeps $40000+ $20000 = $60000 in cash. The reason depreciation is added to after tax earnings when computing cash in the above is that depreciation costs do not involve the spoending of any cash but are just an allocation of some previously spent money. Depreciation to the accountant is a method of cost allocation of an asset. As the example above shows, higher depreciation means lower taxes, but greater cash flow.

In determining total depreciation the original cost, estimated life, and estimated salvage value(value at which asset may be sold at end of life), are generally taken into account. The total depreciation over the life of the asset is COST - SALVAGE VALUE. This is allocated over the life of the asset. There are many ways of doing this. Some involve much depreciation at the beginning of the asset's life and very little at the end while others given equal depreciation every year, etc. The consequences of the different methods of depreciation are many. High depreciation early means greater early cash flow but possibly less later on, etc. The book value of the asset at any time is generally considered to be the original cost - total accumulated depreciation charges.

In the present program the declining balance method of depreciation allocation is used. In this method the yearly depreciation is a constant fixed percentage of the previous year's book value.Given the cost,salvage value, and life in years of an asset this program will construct a simple table containing for each year in the life of the asset the depreciation for that year, the total accumulated depreciation and the book value of the asset. A table for a 3 year lifetime asset costing $10000 with a salvage value of $1000 follows, and with 40% yearly depreciation follows:

YEAR	CURR.DEP.	ACCUM.DEP.	BOOK VALUE
0			10000.00
1	4000.00	4000.00	6000.00
2	2400.00	6400.00	3600.00
3	2600.00	9000.00	1000.00

Note how the last year's depreciation was adjusted to depreciate to the salvage value during the last year.

DEPRDDB

One of the expenses of operating a business is the loss in value of physical property due to factors such as wear and tear, deterioration and decay, obsolescence, etc. Such a loss in value is called depreciation and is generally treated as an expense on a company's income statement. All other things being equal, greater depreciation charged against an asset in a given year will mean lower profits for the corporation. However, lower profits mean lower taxes so the higher depreciation may actually mean that the company is better off cash wise even though it has lower profits. PP CORP. earned $100000 before taxes and depreciation charges and is in the 50% tax bracket. If depreciation is $10000 then the company earns $90000, pays $45000 in taxes and keeps $45000 + $10000 = $55000. If depreciation is $20000 then the company earns $80000, pays $40000 in taxes and keeps $40000+ $20000 = $60000 in cash. The reason depreciation is added to after tax earnings when computing cash in the above is that depreciation costs do not involve the spoending of any cash but are just an allocation of some previously spent money. Depreciation to the accountant is a method of cost allocation of an asset. As the example above shows, higher depreciation means lower taxes, but greater cash flow.

In determining total depreciation the original cost, estimated life, and estimated salvage value(value at which asset may be sold at end of life), are generally taken into account. The total depreciation over the life of the asset is COST - SALVAGE VALUE. This is allocated over the life of the asset. There are many ways of doing this. Some involve much depreciation at the beginning of the asset's life and very little at the end while others given equal depreciation every year, etc. The consequences of the different methods of depreciation are many. High depreciation early means greater early cash flow but possibly less later on, etc. The book value of the asset at any time is generally considered to be the original cost - total accumulated depreciation charges.

In the present program the double declining balance method of depreciation allocation is used. In this method the yearly depreciation is a constant fixed percentage of the previous year's book value.This percentage is double the straight line rate. Given the cost,salvage value, and life in years of an asset this program will construct a simple table containing for each year in the life of the asset the depreciation for that year, the total accumulated depreciation and the book value of the asset. A table for a 3 year lifetime asset costing $10000 with a salvage value of $1000 follows:

YEAR	CURR.DEP.	ACCUM.DEP.	BOOK VALUE
0			10000.00
1	6666.67	6666.67	3333.33
2	2222.22	8888.89	1111.11
3	111.11	9000.00	1000.00

Note how this method gives very high depreciation is early years as opposed to low depreciation charges in later years.

DEPRSF

One of the expenses of operating a business is the loss in value of physical property due to factors such as wear and tear, deterioration and decay, obsolescence, etc. Such a loss in value is called depreciation and is generally treated as an expense on a company's income statement. All other things being equal, greater depreciation charged against an asset in a given year will mean lower profits for the corporation. However, lower profits mean lower taxes so the higher depreciation may actually mean that the company is better off cash wise even though it has lower profits. Depreciation is added to after tax earnings when computing cash flow since since depreciation costs do not involve the spending of any cash but are just an allocation of some previously spent money. Depreciation to the accountant is a method of cost allocation of an asset. Higher depreciation means lower taxes, but greater cash flow.

In determining total depreciation the original cost, estimated life, and estimated salvage value(value at which asset may be sold at end of life), are generally taken into account. In the sinking fund method of computing depreciation the depreciation amounts are considered to be placed in an interest bearing "sinking fund" account. Depreciation is constant from year to year but interest on the fund is considered when computing the change in book value. The annual depreciation amount is chosen is such a way that when considering interest gained on the sinking fund the book value decreases to the salvage value at the end of the item's useful life. The sinking fund method of depreciation is used quite ofter by utilities but is not of much use for other organizations.

As an example of how the program works lets try an example where initial cost is $800,000, salvage value is $150,000, useful lifetime is 5 years and an 8.5% rate is paid on the sinking fund. We run the program and after some initial documentation we see:

```
ENTER COST                  ? <800000><ENTER>
ENTER SALVAGE VALUE         ? <150000><ENTER>
ENTER LIFE IN YEARS         ? <5><ENTER>
ENTER INT.RATE ON SINKING FUND(%) ? <8.5><ENTER>

ANNUAL DEPRECIATION CHARGES = 109698
```

YEAR	BOOK VAL.	FUND INT.	INCR.IN FUND	FUND VAL.
0	800000.00			
1	690302.00	0.00	106698.00	109698.00
2	571281.00	9324.29	119022.00	228719.00
3	442142.00	19441.10	129139.00	357858.00
4	302026.00	30417.90	140115.00	497974.00
5	150001.00	42327.80	152025.00	649999.00

Remember that all computations are done in single precision so large numbers may be a few cents or dollars off.

DEPRSL

One of the expenses of operating a business is the loss in value of physical property due to factors such as wear and tear, deterioration and decay, obsolescence, etc. Such a loss in value is called depreciation and is generally treated as an expense on a company's income statement. All other things being equal, greater depreciation charged against an asset in a given year will mean lower profits for the corporation. However, lower profits mean lower taxes so the higher depreciation may actually mean that the company is better off cash wise even though it has lower profits. PP CORP. earned $100000 before taxes and depreciation charges and is in the 50% tax bracket. If depreciation is $10000 then the company earns $90000, pays $45000 in taxes and keeps $45000 + $10000 = $55000. If depreciation is $20000 then the company earns $80000, pays $40000 in taxes and keeps $40000+ $20000 = $60000 in cash. The reason depreciation is added to after tax earnings when computing cash in the above is that depreciation costs do not involve the spoending of any cash but are just an allocation of some previously spent money. Depreciation to the accountant is a method of cost allocation of an asset. As the example above shows, higher depreciation means lower taxes, but greater cash flow.

In determining total depreciation the original cost, estimated life, and estimated salvage value(value at which asset may be sold at end of life), are generally taken into account. The total depreciation over the life of the asset is COST - SALVAGE VALUE. This is allocated over the life of the asset. There are many ways of doing this. Some involve much depreciation at the beginning of the asset's life and very little at the end while others given equal depreciation every year, etc. The consequences of the different methods of depreciation are many. High depreciation early means greater early cash flow but possibly less later on, etc. The book value of the asset at any time is generally considered to be the original cost - total accumulated depreciation charges.

In the present program a method of depreciation allocation called the straight-line method is used. In the straight-line method of depreciation the yearly depreciation charges for the asset are all equal. Given the cost,salvage value, and life in years of an asset this program will construct a simple table containing for each year in the life of the asset the depreciation for that year, the total accumulated depreciation and the book value of the asset. A table for a 3 year lifetime asset costing $10000 with a salvage value of $1000 follows:

YEAR	CURR.DEP.	ACCUM.DEP.	BOOK VALUE
0			10000.00
1	3000.00	3000.00	7000.00
2	3000.00	6000.00	4000.00
3	3000.00	9000.00	1000.00

The program also gives the option of printing the table on a line printer.

DEPRSY

One of the expenses of operating a business is the loss in value of physical property due to factors such as wear and tear, deterioration and decay, obsolescence, etc. Such a loss in value is called depreciation and is generally treated as an expense on a company's income statement. All other things being equal, greater depreciation charged against an asset in a given year will mean lower profits for the corporation. However, lower profits mean lower taxes so the higher depreciation may actually mean that the company is better off cash wise even though it has lower profits. PP CORP. earned $100000 before taxes and depreciation charges and is in the 50% tax bracket. If depreciation is $10000 then the company earns $90000, pays $45000 in taxes and keeps $45000 + $10000 = $55000. If depreciation is $20000 then the company earns $80000, pays $40000 in taxes and keeps $40000+ $20000 = $60000 in cash. The reason depreciation is added to after tax earnings when computing cash in the above is that depreciation costs do not involve the spoending of any cash but are just an allocation of some previously spent money. Depreciation to the accountant is a method of cost allocation of an asset. As the example above shows, higher depreciation means lower taxes, but greater cash flow.

In determining total depreciation the original cost, estimated life, and estimated salvage value(value at which asset may be sold at end of life), are generally taken into account. The total depreciation over the life of the asset is COST - SALVAGE VALUE. This is allocated over the life of the asset. There are many ways of doing this. Some involve much depreciation at the beginning of the asset's life and very little at the end while others given equal depreciation every year, etc. The consequences of the different methods of depreciation are many. High depreciation early means greater early cash flow but possibly less later on, etc. The book value of the asset at any time is generally considered to be the original cost - total accumulated depreciation charges.

In the present program a the sum-of years digits method of depreciation allocation is used. In this method of depreciation the yearly depreciation charges for an item are greatest early in the life of the asset. Given the cost,salvage value, and life in years of an asset this program will construct a simple table containing for each year in the life of the asset the depreciation for that year, the total accumulated depreciation and the book value of the asset. A table for a 3 year lifetime asset costing $10000 with a salvage value of $1000 follows:

YEAR	CURR.DEP.	ACCUM.DEP.	BOOK VALUE
0			10000.00
1	4500.00	4500.00	5500.00
2	3000.00	7500.00	2500.00
3	1500.00	9000.00	1000.00

The program also gives the option of printing the table on a line printer.

DILANAL

When one publicly traded firm(acquiring firm) decides to merge with another(acquired firm) by offering an exchange of stock the analysis is generally dominated by the exchange ratio (the number of shares the acquiring firm offers for each share of the acquired firm), the price earnings ratios of the two firms(P-E ratio), and the general conditions of the stock market. It is desireable that the earnings per share (EPS) of the acquiring firm not decrease. Such a decrease is called DILUTION. If dilution occurs and the P-E ratio of the merged firm is the same as that of the acquiring firm then the price per share of the stock of the acquiring firm will decrease. This would tend to hurt the acquiring firm's stockholders. More likely, the P-E ratio of the merged firm will be somewhere between that of the acquired and the acquring firm.

In **DILANAL** the dilution occurring in the above type of merger is examined. Our input is the following for each of the two companies involved: earnings after taxes, number of shares outstanding, share price at time of exchange. We also input the exchange ratio. Output consists of the number of new shares, earnings after taxes and EPS for the surviving firm. Let's see how this works in the following case. Company A has offered 2 of its own shares for each share of Company B. Company A has 4500 shares outstanding and last year had earnings after taxes of $105,000. Company B has 10000 shares and last year had earnings of $37,000. Company A stock now sells for $40 per share while Company B stock sells for $10 per share. If Company A buys out company B then what will the EPS of the resulting company be?
We run **DILANAL** and after some initial documentation we see:

```
ANSWER THE FOLLOWING FOR THE ACQUIRING FIRM

EARNINGS AFTER TAXES          ? <105000><ENTER>
SHARES OUTSTANDING            ? <4500><ENTER>
PRICE AT TIME OF EXCHANGE     ? <40><ENTER>

ANSWER THE FOLLOWING FOR THE ACQUIRED FIRM

EARNINGS AFTER TAXES          ? <37000><ENTER>
SHARES OUTSTANDING            ? <10000><ENTER>
PRICE AT TIME OF EXCHANGE     ? <10><ENTER>

INPUT EXCHANGE RATIO          ? <2><ENTER>

                ACQUIRING(BEFORE)     ACQUIRED      ACQUIRING(AFTER)
EAT                   105000.00        37000.00          142000.00
SH. OUTST.              4500.00        10000.00           24500.00
EPS                       23.33            3.70               5.80
P-E RATIO                  1.71            2.70

EXCHANGE RATIO IS 2 ACQUIRING FOR EACH 1 ACQUIRED

ASSUMING NEW P-E RATIO WILL BE BETWEEN 1.71429 AND 2.7027
NEW PER SHARE PRICE WILL BE BETWEEN 9.93586 AND 15.6646
```

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

DISCBAL

Most bank loans are of the discount type. When describing the loan the bank officer(and the loan agreement) state that a certain total amount of payments are required. The total interest charges are also stated in the loan agreement. The proceeds of the loan are computed to be total payments minus total finance charges. An effective interest rate can then be computed and in many states is also stated on the loan. In many cases there are some further mandatory charges. These may include such things as accident and health insurance, credit life insurance, and official fees such as loan origination fees. These extra fees actually raise the cost of the loan to the user but generally banks do not consider it that way. The effective rate, when stated, ignores these other charges.

DISCBAL computes the effective interest rate on a discount loan. The extra charges mentioned above are considered and the answer given by this program is more indicative of the true interest rate than the bank stated rate when these charges exist.

Lets see how the program works with the following example:

> Peter bought a new car for $5559.55. He received a loan from the bank witht the following characteristics: Total amount of repayments = $7296;total prepaid finance charge=$1459.20; total prepaid A & H ins. = $166.35; total prepaid credit life insurance = 110.90. There were 48 payments of $152 each. What was the effective interest rate?

We RUN the program and see:

```
TOTAL REPAYMENTS OF LOAN                ? <7296><ENTER>
TOTAL INTEREST PREPAID                  ? <1459.20><ENTER>
TOTAL A & H INSURANCE CHARGES           ? <166.35><ENTER>
TOTAL CREDIT LIFE INS. CHARGES          ? <110.90><ENTER>
TOTAL OFFICIAL FEES CHARGES             ? <0><ENTER>
MONTHLY LOAN PAYMENT                    ? <152><ENTER>
TOTAL NUMBER OF MONTHLY PMTS            ? <48><ENTER>

PROCEEDS OF LOAN              $5559.55
ANNUAL EFFECTIVE INT. RATE      14.03%
```

The bank would have printed a rate of 11.40% as it would have considered the chareges for insurace as part of the proceeds given to the borrower.

EFFECT

In our economy the time value of money is an important concept. Typically we would rather have money now than 2 years from now (a bird in the hand is worth two in the bush). In order to get money now we are willing to pay an extra fee. This extra fee for borrowing money is interest. The more desperate we are for money the higher the rate of interest we are willing to accept for our loans. Now if we borrow $1000 on January 1, 1970 and we repay $1100 on January 1,1971 then we know that the annual rate of interest was 10%. However most loans involve a periodic payment of some fixed amount over some fixed period, e.g. $100 every month for three years. The total of payments on the loan will always be more than the proceeds of the loan (i.e. the amount you get) since interest rates are positive. The total of the payments - the proceeds of the loan = interest paid. Exactly what is the interest rate the bank is charging me for my loan. The output of the present program is an estimate of this **effective** interest rate.

Let's look at an example:

John borrowed $2214 from ABC Bank to be paid off in 36 equal monthly installments of $75 each. What is John's effective interest rate on his loan?

We run the program and after some initial documentation we see:

```
TOTAL AMOUNT BOROWED          =?_  We enter $2214 <ENTER>

NUMBER OF PAYMENTS YEARLY     =?_  We enter 12 <ENTER>

TOTAL NUMBER OF PAYMENTS      =?_  We enter 36 <ENTER>_

AMOUNT OF EACH PAYMENT        =?_  We enter 75 <ENTER>


EFFECTIVE INTEREST RATE  = 14.2386
TOTAL INTEREST PAID      = 486
```

The output of this loan is but an estimate and seems to be consistently higher than the rate the banks print of their loan applications. A result closer to what the banks themselves get (in fact exactly the same) is gotten by using a good annuity computation program of which one exists in this package. It happens that with annuities the computation of the interest rate is the hardest computation relating to standard computations. The present program does the computation of interest rate very quickly but at the expense of accuracy. This type of trade off occurs in just about all types of business application of quantitative methods.

ENVELOPE

As its name implies, **ENVELOPE** is a program which will print up envelopes on a line printer. In order to use this program reasonably a line printer must be available, preferably a line printer with friction feed. On a tractor feed line printer it would be necessary to have envelopes on tractor forms. This is a menu run program and to give an example of how it operates suppose we wanted to send out a letter to Joan Goldstein,4 Apple Court, West Orange, NJ 07231. The return address is Kimmy Leigh Shenkin, 7321 Golden Rd, Covona, NY 10988. We RUN the program and see:

```
          MENU

1. INPUT RETURN ADDRESS
2. INPUT MAIL TO ADDRESS
3. PRINT ENVELOPE ON SCREEN
4. PRINT ENVELOPE ON PRINTER
5. CLEAR DATA: READY FOR NEW ENVELOPE
6. CHANGE PRINT TABS FOR RETURN ADDRESS OR MAIL TO ADDRESS
7. CHANGE # LINES ON ENVELOPE

MAKE YOUR CHOICE?
```

The envelope is formatted in the following way: The envelope has a total of 22 lines. The return address and the mail to address each have up to 4 lines. The return address starts on the 2nd line of the envelope and in the 2nd column from the left. The mail to address starts on the 9th line and the 30th column to the right. The number of lines in the envelope may be changed via option 7. When we choose this option we see:

```
PRESENT NUMBER OF ENVELOPE LINES = 22

NEW NUMBER OF LINES ? <20><ENTER>
```

The program then returns to the MENU.

There is also an option (option 6) which can alter the column number where either the return address or the mail to address is printed. Option (3) is used too review the envelope on the screen for accuracy. The printing option (option (7)) gives the user the choice of how many envelopes he wants printed. After each envelope is printed the user sees:

```
PRESS ENTER TO CONTINUE?
```

When ENTER is pressed the next envelope is printed. When the last envelope asked for is printed the program returns to the MENU.

Option (5) permits the user to clear both the original and return address and reset the tab and line variables to their defaults

EOQ

A routine problem of most businesses is trying to determine how much inventory to keep in stock. Questions such as the following must be answered: How often should we replenish inventory? When we replenish how much should we reorder? The answers to these questions are not obvious. On the one hand the businessman would like to keep a large inventory. This would guarantee that we could always met demand for orders and reap the consequential profits. On the other hand, holding a large inventory has many costs of its own. Among these are that funds are tied up and interest is being paid on the inventory. Also, holding large inventories means that larger facilities must be provided. This also means more expenses.

Inventory theory has been pretty well studied mathematically. Based upon certain ideal assumptions mathematical inventory theory will give answers to most inventory problems which, if implemented, will at least theoretically result in a lower overall inventory cost and thus a higher overall profit. The model used in this program is one of the simplest used in inventory theory .Before running an example lets define some terms:

A **holding cost** is a cost resulting from storing items in inventory. In our example holding cost will be expressed in dollars per item held per year.

A **reorder cost** is a fixed cost which occurs each time the inventory is replenished, e.g. for paperwork. In our case reorder cost is defined in terms of dollars per reorder.

A **stockout cost** is a cost which occurs when available supply is depleted. A stockout cost may involve loss of the profit from a sale, loss of goodwill, etc. Stockout costs are measured either in units of dollar cost per item short or sometimes in dollar cost per item short per unit of time. In our case we assume that stockouts are totally intolerable so we never allow stockouts.

In our program we assume the following: Demand is known and is uniform throughout the year. There are no stockouts and we can take delivery of ordered items whenever we wish. What the last sentence means is that we can get immediate delivery from our suppliers of anything we order.With the above assumtions, if we have demand of 1200 items per year, a reorder cost of $76 per reorder and a holding cost of $6 per item per year then how much should we reorder at a time and how often? We run the program and see:

```
DEMAND ?_ We press 1200 <ENTER> as demand is measured per year.
REORDER COST ?_ We press 76 <ENTER>
HOLDING COST ?_ We press 6 <ENTER>

TO MINIMIZE TOTAL COSTS OF HOLDING INVENTORY:
          REORDER 174 ITEMS AT A TIME.
THIS MEANS ORDER APPROXIMATELY 7 TIMES PER YEAR
OR ABOUT EVERY 52 DAYS.
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

EPSEST

An estimate of future earnings per share is crucial for many stock evaluation systems. Based upon several current and expected parameters, this program will compute an expected earnings per share for any time in the future. The parameters which the program considers are the following: the number of years into the future for which the estimate is desired, say N, Current sales, expected sales in N years, current profit margin, expected profit margin in N years, current turnover, expected turnover in N years, current leverage, expected leverage in N years, current profit on net worth, expected profit on net worth in N years, current dividend payout ratio, expected dividend payout ratio in N years, book/market ratio, and current earnings per share.

The output is estimated earnings per share N years into the future.

As a sample we shall estimate the EPS 5 years into the future for a company with the following balance sheet-income statement values and expected values 5 years into the future:

Current sales:	$1,000,000
Expected sales:	$3,000,000
Current profit margin:	10%
Expected profit margin:	9.5%
Current turnover:	3.2
Expected turnover:	3.5
Current leverage:	1.1
Expected leverage:	2.3
Current profit on net worth:	.31
Expected profit on net worth:	.23
Dividend payout ratio:	.13
Expected dividend payout ratio:	.20
Book/Market ratio:	.3
Current EPS	$2.00

We RUN the program and after some initial documentation we see:

```
HOW MANY YEARS INTO THE FUTURE SHOULD ESTIMATE BE ? <5><ENTER>
CURRENT SALES                                     ? <1000000><ENTER>
EXPECTED SALES IN 5 YEARS                         ? <3000000><ENTER>
CURRENT PROFIT MARGIN                             ? <.1><ENTER>
EXPECTED PROFIT MARGIN IN 5 YEARS                 ? <.095><ENTER>

- - - - - - - - - - - - - -
- - - -- - - - - - - - - - -

ENTER BOOK/MARKET RATIO                           ? <.3><ENTER>
CURRENT EARNINGS PER SHARE                        ? <2.00><ENTER>

EXPECTED EARNINGS PER SHARE IN 5 YEARS = 10.1
```

EXPVAL

In many business situations the final result may be considered to be some monetary payoff. If the businessman knew for sure what the outcome would be beforehand then he would know whether or not to invest. A possible situation is for there to be several possible outcomes or scenarios with a known (or almost known) monetary payoff for each and with the relative likelihoods of occurrence of each known. In such a situation the expected value yields an 'average payoff'. What this means is that if this situation occurred often then on average the businessman would gain the expected value. One theory of decision making says that all other things being equal the businessman should accept business deals with higher expected values.

In this problem the operator enters the monetary payoff and relative likelihood of each possible payoff. The outputs are the entered items along with some computations and the expected value of the situation.

We shall use the following example or the program: A TEXAS oilman is trying to decide whether to drill a well. Based upon geological considerations he estimates the following:

Payoff	-150,000	-50,000	-10,000	10,000	400,000
Probability	.4	.2	.1	.1	.2

What is the oilman's expected gain?

We RUN **EXPVAL** and after some initial documentation we see:

```
FOR EACH POSSIBLE PAYOFF INPUT PAYOFF AMOUNT AND PROBABILITY.
INPUT 0,0 TO COMPLETE ENTRY AND TO DO COMPUTATION.

ENTRY 1 ENTER PAYOFF,PROBABILITY (0,0 TO END)? <-150000,.4><ENTER>
ENTRY 2 ENTER PAYOFF,PROBABILITY (0,0 TO END)? <-50000,.2><ENTER>
ENTRY 3 ENTER PAYOFF,PROBABILITY (0,0 TO END)? <-10000,.1><ENTER>
ENTRY 4 ENTER PAYOFF,PROBABILITY (0,0 TO END)? <10000,.1><ENTER>
ENTRY 5 ENTER PAYOFF,PROBABILITY (0,0 TO END)? <400000,.2><ENTER>
ENTRY 6 ENTER PAYOFF,PROBABILITY (0,0 TO END)? <0,0><ENTER>
IS LINE PRINTER OUTPUT DESIRED(Y/N)? <Y><ENTER>
ENTRY    VALUE        PROB.        PROB.        ENTRY
                      ENTERED      STANDARD     EXP VAL.
 1   -150000.00        0.400        0.400      -60000.00
 2    -50000.00        0.200        0.200      -10000.00
 3    -10000.00        0.100        0.100       -1000.00
 4     10000.00        0.100        0.100        1000.00
 5    400000.00        0.200        0.200       80000.00
        EXPECTED VALUE EQUALS                    10000.00
```

The sum of the likelihoods oor probabilities entered should equal 1 but if not the progrm takes them as relative likelihoods and computes the PROB. STANDARD column. The oilman's expected profit is $10,000.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

FINRAT

Among the yardsticks the financial analyst and accountant uses to evaluate the performance and financial condition of a firm are certain ratios or indices. These ratios may be used in several ways. First of all, by analyzing comparable ratios for the same firm in a succession of years the analyst may determine certain trends in the firm's financia situation. These may be evaluated as improvements or deteriorations of the firm's condition. For example, if earnings per share(EPS) prove to be increasing by 50% per year the analyst may think one thing while if EPS remains static it might mean something else. Ratios may also be compared among different firms in the same industry. For example, if inventory turnover for one company is substantially greater than that of another similar company then the first company is making better use of its investment in inventory.

There are somewhere between 50 and 100 commonly used financial ratios. Most of these are computed using only data found in the company's balance sheet and income statement. **FINRAT** is a menu driven program which computes and displays 13 commonly used ratios. It should be quite easy to extend the program to more ratios. Statements are fully remarked.

When the program is RUN there is some initial documentation and then we see:

```
          MENU

1. INPUT BALANCE SHEET AND INCOME DATA
2. PRINT OUT BALANCE SHEET AND INCOME DATA AS INPUT
3. EDIT INPUT DATA
4. COMPUTE AND PRINT OUT APPROPRIATE RATIOS
5. STORE COMPANY INPUT DATA ONTO DISKETTE
6. READ COMPANY INPUT DATA FROM DISKETTE
7. STORE COMPANY INPUT DATA ONTO CASSETTE
8. READ COMPANY INPUT DATA FROM CASSETTE
9. CHANGE DATE

INPUT YOUR CHOICE? <1><ENTER>

ENTER THE FOLLOWING DATA

SALES                              ? <46308><ENTER>
```

This follows for 27 income statement and balance sheet numbers. At the end of this the program returns to the MENU. If option 2 is chosen the following type printout appears:

```
DATE IS 09/20/80
1  SALES                           46300.00
2  COST OF GOODS SOLD              39217.00
```

The printout continues for 27 lines in the initial version of FINRAT. The integers to the left of the printout are item numbers used in the EDIT option 3. When option 3 is chosen the 27 lines are again printed, but at a slow rate. Then the user is asked:

```
INPUT ITEM # TO CHANGE (0 TO END)? <2><ENTER>

2  COST OF GOODS SOLD                  39217.00 ?<39852><ENTER>

INPUT ITEM # TO CHANGE (0 TO END)? <0><ENTER>
```

The program returns to the MENU. Option 4 requires data to have previously been input by option 1 or via diskette or cassette. It simply does the computations and prints out a report of the form:

```
DATE IS 09/20/80
EPS                                  0.34
TURNOVER OF OPERATING ASSETS         1.52
```

etc. Note that there are ratios here which generally are written as percentages. This shall not be done and all such ratios are in decimal form. Thus 1.52 could mean 152%.

Option 9 is for changing the date. Date is used for nothing but the printouts, which may be on line printer if desired. Thus the user may use the date change to keep track of the last time the data was modified, or the date of the balance sheet and income staement, even a comment.

Options 5 through 8 relate to storing the input date on diskette or cassette and then recalling the data. When diskettes are used any leagal name may be used as the data file.Thus data for several companies and for several years may be stored on the diskette.

FQEOQ

One of the most important things a businessman can learn to do is manage his inventory efficiently. In this program an optimum order size is computed using the fixed quantity model witht eh following assumptions:

There is a constant ordering cost per order of replenishment stock
Inventory holding cost is constant per unit held/unit time.
No backorders are permitted i.e. replacement stock will always be available.
Order lead time is constan and known.
Demand for the items are constant and uniform through time.
There are no quantity price breaks

The businessman needs the following information to input: lead time, annual demand, ordering cost/order, holding cost/unit/year, and unit purchase price. The output includes optimal order quantity, reorder point, and total annual cost of the inventory system.

As an example suppose H & E COMPUTRONICS has an annual demand for 8000 boxes of 8" diskettes. These diskettes currently cost Computronics $34.99/box. Computronics' supplier always gets the order to Computronics exactly 3 days after the placement of the order. The Computronics Executive Committee estimates that it costs $3.00 to hold a box of diskettes per year and that it costs $17.50 for Computronics to place an order. How many diskettes should Computronics buy at a shot?

We RUN the program and after some initial documentation we see:

```
ENTER ANNUAL DEMAND(IN UNITS)       ? <8000><ENTER>
ENTER REORDER COST(PER ORDER)       ? <17.50><ENTER>
ENTER LEAD TIME(DAYS)               ? <3><ENTER>
ENTER HOLDING COST(PER UNIT/YEAR    ? <3><ENTER>
ENTER PURCHASE PRICE PER ITEM       ? <34.99><ENTER>

OPTIMAL ORDER QUANTITY IS           305.505 UNITS
THIS MEANS ORDER ABOUT 26.1861 TIMES PER YEAR.
     OR ABOUT EVERY 13.9387 DAYS
THE REORDER LEVEL IS 65.7534 UNITS, I.E. WHEN THIS QUANTITY
     IS REACHED REORDER 305.505 UNITS.
TOTAL INVENTORY COST(HOLDING + REORDERING + COST OF ITEMS)
     PER YEAR EQUALS 280836

PRESS ENTER FOR MORE?
```

Of course we are not going to order fractional amounts, but if we round then the results are still alot better than seat of the pants thinking in most cases. Of couse the reorder cost and holding cost had to be estimated by the user. Good initial estimates of these quantities yield good program results.

FQEOQPB

Many businessmen find themselves in the following situation. They purchase from a given supplier and have been doing so for some time. The supplier then runs a sale and offers a discount in unit cost if they buy an extra large quantity. Should the businessman take the quantity price break? The answer is not always easy. On the one hand the unit cost is less. On the other hand, the extra goods cannot usually be sold immediately. It cost money to hold this inventory, i.e. money for storage and for lost opportunity due to the funds tied up. Inventory theory can come to the aid of this businessman and help him to quantify the costs. In this program the quantity price break problem is tackled in the case in which the producer offers price breaks for several fixed quantity purchases, i.e. to get a price break a purchase must be at one of several announced amounts.

The following assumptions are made:

There is a constant ordering cost/order for any number of orders
Inventory holding cost is a constant per unit held/unit time
No backorders are permitted, i.e. stockouts are not allowed.
Demand for the item is constant and uniform through time
Quantity price breaks are offered at several fixed quantities.

With the above assumptions we try to determine which/if any quantity price break should be taken. The following example should clarify the situation.

Computronics buys Diskettes at $36.99 per box. There is demand for 8000 boxes of diskettes per year. It costs $2.00 per year to hold the a box of diskettes in inventory. Due to paperwork,phone calls, etc. it is felt that the cost of any one diskette order is $17.50. Computronics supplier offers 2 price breaks: 1000 boxes @ $34.99 per box and 2000 boxes @ $32.90 per box. How should Computronics order? We RUN the program and see:

```
ENTER ANNUAL DEMAND(IN UNITS)            ? <8000><ENTER>
ENTER REORDER COST(PER ORDER)            ? <17.50><ENTER>
ENTER HOLDING COST(PER UNIT/YEAR         ? <2><ENTER>
ENTER NORMAL PURCHASE PRICE/ITEM         ? <36.99><ENTER>
HOW MANY DIFFERENT LOT SIZES ARE OFFERED? <2><ENTER>
FOR LOT SIZE # 1
     LOT SIZE QUANTITY    ? <1000><ENTER>
     PRICE/UNIT           ? <34.99><ENTER>
FOR LOT SIZE #2
     LOT SIZE QUANTITY    ? <2000><ENTER>
     PRICE/UNIT           ? <32.90><ENTER>

#    SIZE       PRICE/UNIT          TOTAL COST
0     374.2       36.99              296668.00
1    1000.0       34.99              281060.00
2    2000.0       32.90              265270.00
MINIMUM COST IS 265270 ATTAINED BY CHOOSING LOT SIZE 2000.
```

FQEOWSH

As many businessmen have learned the hard way, it does not always pay to have everything ordered in stock. This is because to have stock of all items in large quantities means that the inventory holding cost is quite large. This additional cost of holding inventory may be more than the extra earnings generated by always having items in stock. In this program, if the businessman knows the cost/unit/year for holding an item and the cost/unit/year of not having an item in stock, he may be able to compute just how many items he should order at a shot. In particular, we make the following assumptions:

There is a constant cost in placing a single order(reorder cost)
Inventory holding cost is constant/unit held/unit time
Stockouts are permitted(backorders allowed)
Demand for the items is constant and uniform through time.
There are no quantity price breaks.
Production runs(or reorders) must be planned in advance.

The output includes optimal reorder quantity, the optimal time between reorders, and the expected maximum inventory quantity per reorder period.

As an example. Let's look at H & E Computronics. Computronics needs 8000 boxes of 8" diskettes per year and pays $34.99/box. It is estimated that it costs Computronics $3.00 to hold a box of diskettes for a year while it costs $4.50 per box short/year. Computronics feels that it costs $17.50 to place any size order for diskettes. How much should Computronics buy at a shot?

We RUN the program and after some initial documentation we see:

```
ENTER ANNUAL DEMAND(IN UNITS)       ? <8000><ENTER>
ENTER REORDER COST(PER ORDER)       ? <17.50><ENTER>
ENTER HOLDING COST(PER UNIT/YEAR    ? <3><ENTER>
ENTER SHORTAGE COST(PER UNIT/YEAR   ? <4.50><ENTER>

OPTIMAL ORDER QUANTITY IS                  394.405 UNITS
THE OPTIMAL TIME INTERVAL BETWEEN REORDERS  =  .0493007
THE EXPECTED MAXIMUM INVENTORY LEVEL FOR EACH ORDER PERIOD
     EQUALS 236.643 UNITS

PRESS ENTER FOR MORE?
```

Of course we will not order a part of a unit but if we round then the ordering decisiuon will probably be much better than a seat of the pants method.

FUPRINF

How many dollars will today's $4,000 cost in 5 years, 8 years, or 10 years? If my salary increases at a 6% yearly rate then how much will I be making in 10 years if I make $15,000 now? These are some questions people ask.**FUPRINF** will take the current cost of an item and the expected yearly per cent price increase in the item and print a table giving the cost of the item over a user selected number of years where the cost is in dollars.

To see how the program operates we shall estimate the price of today's $4,000 automobile over each of the next 10 years assuming a yearly price increase of 13%. To do this we RUN the program and after some initial documentation we see:

```
ENTER PRESENT COST OF THE ITEM              ? <4000><ENTER>
HOW MANY YEARS INTO THE FUTURE ARE DESIRED?<10><ENTER>
WHAT IS THE ESTIMATED YEARLY % COST INCR. ?<13><ENTER>
IS LINEPRINTER OUTPUT DESIRED(Y/N)? <Y><ENTER>
```

We then get the following on the printer with an almost identical CRT printout:

```
PRICE ESTIMATION @  13% PRICE INFLATION
YEAR      COST       % INCR. ON ORIG. COST

  0      $4000.00          0.00
  1      $4520.00         13.00
  2      $5107.60         27.69
  3      $5771.59         44.29
  4      $6521.89         63.05
  5      $7369.74         84.24
  6      $8327.80        108.20
  7      $9410.41        135.26
  8     $10633.80        165.84
  9     $12016.10        200.40
 10     $13578.20        239.46
```

Thus in the 10 years the price almost tripled. It may be distressing to note that if someone makes $20,000 now and has a 10% salary increase per year then after 10 years the program shows a salary of $51874.80. This is less than 4 times the cost of the car while now the $20,000 covers five times the cost of the car. The few extra percentage points (13% vs 10%) when inflation is being considered.

FVAL

FVAL finds the future value of an investment for a stated period of time at a given interest rate. It is assumed that the value of this investment is compounded periodically. A question which one might ask is "What does future value mean?". The answer is quite simple. Suppose you place $100 in a vault and can only take it out in 10 years without getting anything extra. Then the future value of your $100 in 10 years is $100. Suppose, however, that you know that there will be $150 waiting for you in that vault after those 10 years? Then we would say that the future value of that $100 in 10 years equals $150. Thus the future value of a money investment after a fixed amount of time is the amount of money that the investment will be worth after that amount of time.

Some typical questions which might be solved by being able to compute future values are the following:

1. If we put $1000 in a special trust account for our child at birth and if the account pays 8% annually compounded quarterly then how much will be in the account at age 18?

2. Joe Loan Shark loans me $100 at 30% daily interest rate, compounded daily. How much will I owe Joe after 7 days?

How would we use FVAL to solve these problems? We try running the program and after some initial documentation we see:

IS INTEREST RATE QUOTED ANNUALLY WITH COMPOUNDING(Y/N)? _ We answer Y<ENTER> for question 1. The computer needs an interest rate per compounding period. This option computes the quarterly rate as 1/4 the annual rate,e.g..For problem 1 we need a quarterly(periodic) rate.

ANNUAL INTEREST RATE IN % - We enter 8 <ENTER>
NUMBER OF COMPOUNDINGS ANNUALLY ?_ We enter 4 <ENTER>

INTEREST RATE PER PERIOD = 2%

AMOUNT OF INVESTMENT? - We enter $1000
INTEREST RATE PER PERIOD:? - We just press <ENTER> and 2% is entered automatically
NUMBER OF PERIODS:? We enter 72 as this equals quaters in 18 years.

FUTURE VALUE = 4161.15

PRESS 1 FOR MORE TRIALS, 0 TO END ?_ We press 1 <ENTER> and then similarly solve the second problem. Note that in this second problem interest rates are not quoted annually. By the way the answer to the second problem equals $627.49 which tells you who not to borrow from.

INSFILE

INSFILE is a very simple program which permits an individual or small company to store data about the portfolio of all insurance policies held. For each policy the following data is stored: type of insurance, policy description, expiration date, name of insurance company, underwriter, policy number, face value amount, and annual premium. Some of these items must be entered in some user-defined coded form since the display is formatted to fit on one line of the CRT or,if printer desired, an 80 column printer. Thus, for instance, type of insurance should be no more than 4 characters long, description no longer than 10 characters, etc. Use of the system will show the user exactly how many characters can be entered. By the way, the machine will store any number of characters for any variable. It's just that not all of them will be displayed on the CRT or printer. When we run the program we first see some documentation and then:

```
          MAIN MENU

1. ADD TO FILE OR BEGIN NEW FILE
2. READ FILE IN FROM TAPE
3. WRITE FILE OUT ONTO TAPE
4. READ FILE IN FROM DISK
5. WRITE FILE OUT ONTO DISK
6. EDIT FILE
7. LIST CONTENTS, COMPLETE OR BY TYPE
8. ROUTINE ACTUALLY DELETING RECORD FROM FILE

ENTER YOUR CHOICE? <1><ENTER>

                              TYPE INSUR.       ?<HOUS><ENTER>
                              POL. DESCRIPTION ?<FLOOD><ENTER>
                              DATE EXPIRES      ?<12/31/81><ENTER>
                              COMPANY WRITING  ?<AETN><ENTER>
                              UNDERWRITER       ?<KOYLE><ENTER>
                              POLICY NUMBER     ?<FLH675123><ENTER>
          0.00                   FACE AMOUNT    ?<25000><ENTER>
          0.00                   ANNUAL PREMIUM?<125><ENTER>

TYPE  DESCRIPT.  EXPIRES  CO.  UNDWRT   POL #      AMT.    PREM.
HOUS  FLOOD      12/31/81 AET  KOYLE  FLH675123   25000    125
ENTER Y IF OK,N IF ERROR,A TO ABORT? <Y><ENTER>
DO YOU DESIRE MORE ENTRIES(Y/N)?
```

An A above returns the user to the main MENU, with this last entry ignored. An N causes each of the questions for this policy to be repeated. The present value of the answer to each question is displayed towards the left, the category in the center with room for new input towards the right. The user then enters the new value and presses <ENTER> or,for no change just presses enter. For example, for the first question we see:

```
TYPE  DESCRIPT.  EXPIRES  CO.  UNDWRT  POL #       AMT.     PREM.
HOUS  FLOOD     12/31/81  AET  KOYLE  FLH675123     25000   125
ENTER Y IF OK,N IF ERROR,A TO ABORT?<N><ENTER>
HOUS                      TYPE INSUR.      ?<ENTER>
                (The above <ENTER> means no change)
FLOOD                     POL. DESCRIPTION ?<ENTER> (no change)
12/31/81                  DATE EXPIRES     ?<01/31/82><ENTER>
                                        (so new value)
```

and so forth until the ANNUAL PREMIUM question is asked again. Then the single line display and the Y,N,A question come up again. When Y is the answer this policy is added to the policy file.

Option (6) from the MAIN MENU, the EDIT FILE option displays the file in groups of 12 policies along with an item number for each policy. The item number is entered to bring a record into the edit mode. The edit mode itself is just like the additional input after an N answer to the Y,N,A question at the end of the original input section of the program as demonstrated above. One fact worth noting. If policy type is changed to ZZZZ then that policy is to be deleted. However, to actually delete the policy option (8) (ROUTINE ACTUALLY DELETING RECORD FROM FILE) must be chosen.

Option (7),the LIST CONTENTS option, produces either a complete list of the file or,if desired, a complete list of any policies of a particular type. Total policy face values and annual premiums are also printed. The list may be printed on CRT and line printer. Fo example we may see if we choose the LIST BY TYPE option appearing after choosing option (7) and enter type AUTO we see on the printer(if available):

```
                        LIST OF TYPE AUTO
TYPE   DESCRIPTION  EXPIRES   COMP. UNDWRT   POL #     FACE AMT.   PREMIUM
AUTO   VOLKS        10/22/80  ALLS  ALLST  033 134430    50000        500
AUTO   MALI         10/22/80  ALLS  ALLST  034 123411   100000        635

TOTAL FACE VALUE AMOUNT :  $150000.00
TOTAL PREMIUM AMOUNT    :     $379.00
```

We return to the MAIN MENU

Options (2),(3),(4) and (5) involve saving the file on cassette or diskette and recalling it. The disk options require the entry of a file name, e.g. INDAT. There can be different names and thus many different files on the same diskette. When saving a file onto diskette some time after the file has been read into the memory the following will appear:

```
      PRESENT DISK FILE IS INDAT   NEW FILE? -
or    PRESENT DISK FILE IS NEW FILE?
```

In the 1st case just press <ENTER> to save under the INDAT name. In the 2nd case, we must enter a new file name.

INVENT2

INVENT2 is an in-memory inventory control system. What in-memory means in this case is that all manipulations involving the master inventory file are performed while the entire file lies in the main computer memory. The program permits the master to be stored on either cassette or diskette and to be recalled. But we cannot work with just part of the master in memory and part on diskette. In memory systems have two general properties. First, an in memory system can generally work more quickly than a disk based system (once the master is load into memory) because main memory is much faster than disk memory. Second, we are limited by amount of main memory (e.g. 16k, 32K, etc.) and thus an in-memory system has a smaller capacity than a disk-based system. In the case of the present program we have limited ourselves to a maximum of 50 different inventory item numbers. This number is about the capacity in a system with 16K of free memory. With more memory, more item numbers may be entered onto the master.

We shall review what this inventory system can do by going over the 11 items in the main menu of the program. These are the following:

```
1.ADD ITEMS TO INVENTORY FILE(OR START FILE)
2.INVENTORY LIST(ENTIRE OR BY CLASS OR BY VENDOR)
3.PROCESSING CHANGES OF INVENTORY AMOUNTS
4.READ INVENTORY FILE FROM CASSETTE
5.READ INVENTORY FILE IN FROM DISKETTE
6.WRITE INVENTORY FILE ONTO CASSETTE
7.WRITE INVENTORY FILE ONTO DISKETTE
8.UPDATE TO BEGIN NEW INVENTORY PERIOD
9.ACTIVITY REPORT
10.EDITING INVENTORY  FILE
11.END THIS PROGRAM
```

1. This option permits us to add a new item to the inventory master list or to start a new master. For each item we have the following items to input:

ITEM NO : Can be any sequence of alphanumeric characters.It's better if kept to 7 characters or less for report formatting.

DESCRIPTION: 20 characters or less of description on the item.
VENDOR NO. : Sequence of numbers or characters. Items for any single vendor may be listed individually.
QUANTITY ON HAND: Single precision value indicating the amount of the item in question is in stock.
LAST UNIT COST: The unit cost of this item the last time a shipment was delivered.
SELLING PRICE: The unit price at which the item is sold.

AVERAGE COST: This number times the quantity on hand should give the cost value of inventory. If inventory amounts are only added and subtracted using option 3 from the menu then AVERAGE COST is automatically updated by the program.

REORDER POINT : The quantity level such that having this number or fewer in stock will mean that this item should be reordered.

MAXIMUM INVENTORY:The inventory quantity which means an overstocked condition exists.

ORDERS THIS INVENTORY PERIOD: The quantity of this item which has been sold in the present inventory accounting period, be it day, week, month.

ORDERS LAST INVENTORY PERIOD: As above but for the previous inventory accounting period.

ORDERS 2 INV PERIODS AGO: As above but one period previous.

2. This option permits the either a complete inventory list, or a list of all items from a specified vendor, or a list of all items below their reorder levels. This last report is a very useful report.

3. This is the option via which amounts of any item may be added to or subtracted from inventory. When an item is sold this option is used and when the item is delivered this option is used. Though the editing option 8 can alter inventory amounts this option must be used if average cost and total orders for this inventory period are to be kept up to date.

4.-7. These are the options to read a file into memory and to save it back on cassette or diskette storage. Remember, a master file when read into memory and changed in any way must be saved back onto cassette or diskette or the changes will be lost.

8. At the start of a new inventory accounting period the number of orders for the present period is moved to the number for last period, the number for last period is moved to the number for 2 periods ago and the number for this period is zeroed.

9. This option is a useful report which summarizes for each item the amount on hand, the number of order for this period and the number of orders for the last inventory period.

10. This option permits any data about any item to be edited. Items are referenced by item number.

11. This option asks whether the user really wants to end the program. This is so a master not resaved on cassette or diskette can still be saved.

The best way to learn about the system is to use it.This system should be useful for those with limited numbers of inventory items.

INVOICE

INVOICE is a program which generates invoices on a line printer and the CRT. In any one run up to 50 invoices may be generated. When invoicing is completed a transactions report may be printed out. The easiest way to learn how to use it is to try to generate an invoice. We RUN the program and after some initial documentation we see:

```
IS PRINTER BEING USED(Y/N)? <Y><ENTER>
PREPARE INVOICE IN PRINTER.
PRESS ENTER TO CONTINUE? <ENTER>

ENTER NUMBER OF CUSTOMER TO BE INVOICED(-1 TO END)? <34><ENTER>
CUSTOMER NAME ? <FOLKSY MUSIC INC><ENTER>
2ND LINE NAME ? <ATT. MR. JOHN JONES><ENTER>
STREET ADDRESS? <4 SADORE LANE><ENTER>
CITY,STATE,ZIP? <SPRING VALLEY,NY,10977><ENTER>
PRESS 1 TO INVOICE,2 TO ABORT? <1><ENTER>

INVOICE NUMBER ? <120><ENTER>
DATE OF INVOICE IN FORM MM/DD/YY? <09/17/80><ENTER>
CUSTOMER ORDER NUMBER? <235T><ENTER>
SHIPPED VIA          ? <UPS><ENTER>
TERMS               ? <NET 30><ENTER>

                                          **********
LINE 1 ITEM NO. <12><ENTER>
   ITEM DESCRIPTION <TABLE TOP LIGHT><ENTER>  (printing limit above)
QUANTITY ORDERED,UNIT PRICE ? <5,54.30><ENTER>

NUMBER OF LINES PRINTED SO FAR = 1
ITEM NO.    DESCRIPTION    QUANT.     UNIT PR.    EXTENDED PR.
12        TABLE TOP LIGH       5        54.30        271.50
ENTER Y IF OK,N IF ERROR? <Y><ENTER>
MORE ITEMS ON INVOICE(Y/N)? <Y><ENTER>

                                          **********
LINE 2 ITEM NO. <234><ENTER>
     TABLE LIGHT BRACE
QUANTITY ORDERED,UNIT PRICE ? 5,29

NUMBER OF LINES PRINTED SO FAR = 2
ITEM NO.    DESCRIPTION    QUANT.     UNIT PR.     EXTENDED PR.
234       TABLE LIGHT BR       5        29.00         145.00
ENTER Y IF OK, N IF ERROR? <Y><ENTER>
MORE ITEMS ON INVOICE(Y/N)? <N><ENTER>

TAXABLE(Y/N)? <Y><ENTER>
TAX RATE(%)? <4><ENTER>
```

SHIPPING ? <5.00><ENTER>

Now we see on the screen:

INVOICE NUMBER 120 DATE:09/17/80

ITEM NO.	DESCRIPTION	QUANT.	UNIT PR.	EXTENDED PR.
12	TABLE TOP LIGH	5	54.30	271.50
234	TABLE LIGHT BR	5	29.00	145.00
	TOTAL	10		416.50
	SALES TAX			16.66
	SHIPPING			5.00
	PAY THIS AMOUNT			$438.16

ENTER Y IF OK, N IF ERROR <Y><ENTER>

The invoice is now printed on the line printer. The limitation to 10 lines is only so we can view the invoice on the CRT. The actual invoice forms used could take 18 entries but the user must modify the line limitation put into the program.

DO YOU WANT MORE INVOICES(Y/N) <Y><ENTER>
PREPARE INVOICE IN PRINTER. ETC.

WHEN WE FINALLY ANSWER NO MORE INVOICES AN INVOICE TRANSACTIONS REPORT IS PRINTED. A TYPICAL LINE IN THIS REPORT MIGHT LOOK LIKE:

TRANSACTIONS REPORT

CUST.NO.	DATE	INVOICE #	TYPE	AMOUNT	OTHER CH.
120	09/13/80	120	+	438.16	0

ETC. Company totals are printed at the end of the report.

LABEL1

LABEL1 prints shipping labels on standard 1 inch wide tractor feed label stock. The labels are especially designed for large orders consisting of many boxes. Each label has space for a name, a company name if desired, a street adress, and a city state zip. There is also a query for purchase order number. The first line of each label prints out, e.g. 2 OF 3 P/O #RT65 etc. the easiest way to see how the system operates is to try a simple example, which involves sending an order of 3 boxes to H & E COMPUTRONICS at 50 N. Pascack Rd., Spring Valley, N.Y. 10977 corresponding to COMPUTRONICS purchase order number 3P0897.

We run the program and after some initial documentation we see:

```
HOW MANY LABELS DO YOU NEED? <3><ENTER>
ENTER PURCHASE ORDER NUMBER_ <3P0897><ENTER>
INPUT NAME                  _ <SOFTWARE MANAGER><ENTER>
INPUT COMPANY(ENTER IF NONE)_<H & E COMPUTRONICS><ENTER>
INPUT STREET ADDRESS        _<50 N. PASCACK RD.>
INPUT CITY,STATE,ZIP        _<SPRING VALLEY, NY  10977><ENTER>

PRESS ENTER WHEN PRINTER IS READY?<ENTER>
DO YOU WANT A MASK(Y/N)? <N><ENTER> (A mask is used to line up labels.)

     1 of 3 P/O #3P0897
SOFTWARE MANAGER
H & E COMPUTRONICS
50 N. PASCACK RD.
SPRING VALLEY,NY,10977

     1 of 3 P/O #3P0897
SOFTWARE MANAGER
H & E COMPUTRONICS
50 N. PASCACK RD.
SPRING VALLEY,NY,10977

     1 of 3 P/O #3P0897
SOFTWARE MANAGER
H & E COMPUTRONICS
50 N. PASCACK RD.
SPRING VALLEY,NY,10977
```

LABEL2

LABEL2 is a repetitive label making program. For practical purposes the program will cause any number of identical labels to be printed on standard one across 1" tractor mailing labels. On each label there is provision of from 1 to 5 printed lines. Though at first thought this program seems to be ideal for items like return address mailing labels it may also be used for pricing labels, special telephone number labels (e.g. the number of the security department at your place of work), etc. Let's use the program with the following example:

It is desired to print 10 labels, each of which has printed:

H & E COMPUTRONICS
SECURITY DIVISION
914-425-1535
CALL FOR EMERGENCIES

We run the program and after some initial documentation we see:

```
HOW MANY LABELS ARE DESIRED? <10><ENTER>

FIRST LABEL LINE       <H & E COMPUTRONICS><ENTER>
SECOND LABEL LINE      <SECURITY DIVISION><ENTER>
THIRD LABEL LINE       <914-425-1535><ENTER>
FOURTH LABEL LINE      <CALL FOR EMERGENCIES><ENTER>
FIFTH LABEL LINE       <ENTER>

PRESS ENTER WHEN PRINTER READY? <ENTER>

DO YOU WANT A PRINT MASK(Y/N)? <Y><ENTER>

**************************
**************************     (This is used to line up labels in printer
**************************      and is not printed on screen.)
**************************
**************************

DO YOU WANT A PRINT MASK(Y/N)? <N><ENTER>

H & E COMPUTRONICS
SECURITY DIVIDSION
914-425-1535
CALL FOR EMERGENCIES
```

10 of these are printed and then the user has the option of inputting data for more labels.

LEASEINT

A lease is a contract by a lessee to make a series of payments to a lessor for the use of an asset. The lessee obtains most of the benefits associated with ownership of the asset but the lessor retains title to it until the end of the lease where several things may occur depending upon the particular leasing contract. For example, the lessee may pay some additional amount to gain ownership of the asset or the lessor gains complete control of the asset or perhaps the lessee can renew the lease. In the past a particular advantage of a lease for business was that the lease obligations did not have to be mentioned on the balance sheet. Accounting rulings have made this advantage somewhat nonexistant. Another advantage is that certain leases may be structured so that money is spent by the lessee in the future rather than in the present. This can occur if the lessee must buy the asset at the end of the lease term but must pay a considerable portion of the purchase price at that time.

Though the LEASEINT program is able to analyze quite a few types of loans from an interest standpoint it was designed to determine the interest rate on the so-called open-end auto lease. This method of financing a car works in the following way:

The lessor(leasing company) actually buys the car but the lessee gets use of the car for the term of the lease. In return for this the lessee gives the lessor a down payment, a fixed amount every month, and a buy-back amount at the end of the lease period. The buyback-amount is generally a mandatory figure that the lessee must pay to get title to the car. However, the buyback is set low enough so the car is almost certainly worth more than this amount. Thus if the lessee does not wish to exercise the buyback he may sell the automobile for more than the buyback amount, give the buyback amount to the leasing company and pocket the difference.

Open end leases may have different terms, different down payments, different buyback amounts, different monthly payments, etc. The present program will compute the annual interest rate for a lease in the case where all monthly payments are the same except the down payment and the buyback at the end of the lease. The payments need not even start one month after the lease begins but can begin 2,3,4, etc. months after the beginning.

Input for each lease is the cash cost of the item, the down payment, the monthly payment, the number of monthly payments, the month of the first payment, and the buyback amount. The output is a monthly interest rate, an annual rate(simp) gotten by multiplying the monthly rate by 12 and an annual rate(eff) gotten by compounding the monthly rate. Most institutions seem to use the annual rate(eff) when quoting interest rates.

Simple bank loan rates may be analyzed by having the proceeds correspond to cash cost,a down payment of 0 and a buyback of 0.

LETWRT

This is a mini text editor. The use of LETWRT permits one to write some text, add lines to it, delete lines from it, change it, save it on cassette or diskette and recall from same. Text may also be printed on the line printer. This is a line oriented text editor. Only entire lines may be inserted, deleted or changed. An idea of what the text editor may do is given by its main menu below:(For illustration user input enclosed in< >)

MENU

1.DISPLAY TEXTFILE WITH LINE NUMBERS
2.PRINT TEXTFILE WITHOUT NUMBERS
3.SAVE TEXT FILE ON CASSETTE
4.SAVE A TEXTFILE ON DISKETTE
5.READ A TEXT FILE FROM CASSETTE
6.READ TEXTFILE FROM DISKETTE
7.CHANGE A LINE
8.DELETE A LINE
9.INSERT A LINE
10.ADD TO OR START TEXT FILE
11.PRINT FILE OUT ONTO LINE PRINTER
12.MERGE WITH MAILPAC

WHEN WE USE OPTION 10 WE GET ON THE SCREEN:

```
INPUT LINES END WITH ENTER. FOR INDENTED LINES OR LINES
CONTAINING COMMAS OR COLONS BEGIN WITH A ".
INPUT @ TO RETURN TO MENU
1.? <THE COW JUMPED OVER THE MOON.><ENTER>
2.? <THE DISH RAN AWAY WITH THE SPOON><ENTER>
3.? <"PRINT A,B,C"><ENTER>
4.? <@><ENTER>
```

Remember,the < > are not used in the actual program. After the @ in line 4. we return to the MENU. Note how line 3 was entered within quotes since it had commas in it. After we return to the MENU we may display the file (option 1) , print the file on the line printer (option 2), change the file(options 7-10), or save the file on cassette or diskette. There is no automatic saving of text after additions or editing. Therefore, if a user has a new file or has edited an old file he must save that file (options 3 or 4) before he turns the computer off or the changes and additions will be lost.

The insert a line option(option 9) may cause some confusion. The option asks which line we want to insert. That line number will already exist. The line number mentioned and all following lines in the file get one number higher and we do our insertion at the desired line.

Option 12. MERGE WITH MAILPAC is a very special option. It requires a line printer to use but permits us to generate the same letter for each name on any MAILPAC file. To use this option there must first be a LETWRT file in memory, either one just generated or read into memory from storage. We will suppose that the desired file is the above which isn't a letter but will do just as well for our purpose. When we pick option 12. we see:

```
FILE TO BE READ IN FROM DISKETTE

ENTER NAME OF FILE TO BE READ? <NAMES1><ENTER> (A stored MAILPAC file is
                                                 desired.)
 29 NAMES TO BE READ
PREPARE PRINTER FOR LETTER
PRESS ENTER TO CONTINUE? <ENTER>

1244
PETER SHENKIN
H & E COMPUTRONICS
50 N PASCACK ROAD
SPRING VALLEY NY  10977

NAME 1 OF 29

PRESS 1 TO PRINT LETTER FOR NAME, 0 TO SKIP? <1><ENTER>
```

(The following on the printer)

```
1244
PETER  SHENKIN
H & E COMPUTRONICS
50 N PASCACK ROAD
SPRING VALLEY  NY  10977

THE COW JUMPED OVER THE MOON
THE DISH RAN AWAY WITH THE SPOON
PRINT A,B,C
```

We now automatically skip lines until a new page and display the 2nd name on screen, ready to print. This goes on until the entire name file is traversed. Thus a letter may be printed for any desired name on the list.

When the last name is printed the program returns to the main menu.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

LOANAFFD

One common method of determining whether to give a mortgage to an applicant is to set a fixed percentage, say 25%, and to deny the mortgage if the total monthly payment would be more than that fixed percentage of gross monthly income. Monthly payments to the bank include payments for taxes and(sometimes) insurance. Thus the total monthly payment includes more than just interest and amortization of the mortgage loan.

LOANAFFD gives the maximum mortgage which would be granted to an applicant with a certain monthly gross income. Input required includes gross monthly income, the fixed percentage mentioned above,the percentage of the total monthly payment which is expected to be used for taxes and insurance, mortgage interest rate, and term of mortgage. The following example shows how the program works:

Dr. Davies has monthly gross income of $4500. The bank requires that no more than 30% of gross monthly income be used for monthly payments. In general, about 33% of the total payment goes for taxes and insurance in this region. 30 year mortgages are now costing 12.5 %. What is the largest mortgage which might be granted to Dr. Davies?

To solve the above problem we RUN LOANAFFD and after some initial documentation we see:

```
ENTER APPLICANT'S GROSS MONTHLY INCOME   ? <4500><ENTER>
ENTER FIXED PER CENT                     ? <30><ENTER>
ENTER TAX-INS % OF TOTAL PAYMENT         ? <33><ENTER>
ENTER ANNUAL MORTGAGE INT RATE(%)        ? <12.5><ENTER>
ENTER LENGTH OF MORTGAGE(YEARS)          ? <30><ENTER>

MAXIMUM MONTHLY PAYMENY AMOUNT IS       $1350
CONTAINING INSURANCE AND TAXES OF      $445.50
MAXIMUM MONTHLY PAYMENT FOR MORTGAGE IS  $904.50

MAXIMUM MORTGAGE LOAN AVAILABLE IS   $84750

PRESS ENTER FOR MORE COMPUTATION ? (If we press <ENTER> we can do another
                                    problem of this type.)
```

It is interesting what higher interest rates do to the maximum mortgage amount allowable. If we change the mortgage interest rate above to 9.5% but leave everything else the same we discover that the maximum monthly payment and the maximum monthly payment for mortgage are the same. Now, however, they support a mortgage of $107569.

Beware of the last digit in any of the answers, especially with large number, since all computations are done in single precision.

LOANPAY

This program is a very simple loan program. In short, what LOANPAY does is permit the user to compute the monthly payment on a bank loan if he knows the initial amount received from the loan, the number of monthly payments during the lifetime of the loan, and the nominal interest rate at which the loan is made. There is an option in the program which allows the user to compare loan payment amounts when pnly the interest rates change.

Lets see how the program works.

Peter is borrowing $2214 from the bank and will repay the loan in 36 equal monthly installments. When Peter asks his friendly banker what the interest rate is, the banker says that it is 13.38%. How much should Peter's monthly installment be?

We run MODEBTPA and after some initial documentation we see:

WHAT IS THE AMOUNT OF THE LOAN?_ We press 2214 <ENTER>

HOW MANY PAYMENTS ARE NEEDED?_ We press 36 <ENTER>

WHAT IS THE NOMINAL INTEREST RATE(%)?_ We press 13.38 <ENTER>

MONTHLY PAYMENT EQUALS $75
PRESS 0 TO END,1 FOR NEW LOAN, 2 FOR NEW INTEREST RATE?_ Peter believes interest rates will soon go down to 10% and he wonders if he should wait to get his loan. What would the payment be if the interest rate was 10%. To find this out Peter presses 2 <ENTER> and sees:

WHAT IS THE NOMINAL INTEREST RATE(%)?_ We press 10 <ENTER>

MONTHLY PAYMENT EQUALS $71.44
PRESS 0 TO END,1 FOR NEW LOAN, 2 FOR NEW INTEREST RATE?_
If we now press 1 <ENTER> we can analyze different loan terms.

With the new information Peter better knows whether he should wait for the lower interest rates. It is interesting to see that if Peter had gotten his loan at 24% annual rate then the payments would have been 86.86 monthly while at 50% they would have been 119.81 monthly. Notice how higher interest rates increase payments considerably.

MAILPAC

MAILPAC as its name implies is a mailing list program. It is a rather complete in memory program. The names may be stored on either cassette or diskette and may be recalled from same. In the case of diskette the user is free to name his file and thus several different mailing lists may be stored on one diskette. On cassette just use different cassettes for different files. The number of names which the system will handle depends upon the memory of the system. On a 16K level II about 100 names should fit (no disk drives here) while on a 48K system about 500 names should be handled. The dimensioning on the distributed program is for 99 names but this may easily be increased by the user in case of a more than minimal system.

The following MENU from the program shows what is available:

```
TO BUILD OR ADD TO A FILE TYPE 1
TO SEE THE ENTIRE FILE TYPE 2
TO SEE AN INDIVIDUAL NAME TYPE 3
TO MAKE CORRECTIONS TYPE 4
TO SAVE THE CURRENT FILE ON TAPE TYPE 5
TO SAVE THE CURRENT FILE ON DISKETTE TYPE 6
TO INPUT A FILE FROM TAPE TYPE 7
TO INPUT A FILE FROM DISKETTE TYPE 8
TO PRINT THE ENTIRE FILE(LINE PRINTER) TYPE 9
TO PRINT ALPHABETIZED LIST(LINE PRINTER) TYPE 10
TO PRINT AN INDIVIDUAL NAME(LINE PRINTER TYPE 11
```

To begin making up a mailing list option 1 must be chosen. Put the names and other information in one after another until all names are entered. No field may have any commas in it. This applies in particular to the city,state, zip field which should be written as city state zip (no commas). One field which may stump some people is the field relating to MEMBERSHIP NUMBER. This can really be anything, but when the program was first used COMPUTRONICS used the membership number to keep track of subscriptions.

Once we return to the MENU from the BUILD option we can do anything else we wish but it is highly desirable to SAVE the file we have built up. This is done via options 5 or 6. Remember, this is an in memory system. What this means is that all data must be read into memory from external storage before any data can be modified and all data must be read back onto the cassette or diskette before any changes can be stored. Thus whenever you make any change remember to SAVE. When alot of names are being edited or added to the list the question of exactly how often to save comes up. The more often the edited file is saved the less data is lost in case the system goes down or the operator makes a grievous error. However, saving the file many times takes alot of time and may delay the operation. It is up to the operator to decide this question. Some people are cautious (SAVE alot) and some are adventuresome (SAVE few times). One thing for certain is to be sure you save before turning the machine off.

Option 2 allows one to see the entire file presently in memory. (Naturally, this file was brought in from cassette or diskette earlier.) The file scrolls down the screen rather rapidly but may be stopped at any point by pressing <SHIFT> @ (HOLD on the MOD II). The names scroll in the order ni which they were put into the list.

Option 3 allows to user to examine the data for any individual name. The recall is keyed on last name. The last name must be keyed in exactly as it was entered in the file. If there are duplicate last names this option will find the first of these. There is a duplicate name option which will allow others with this same last name to be displayed.

Option 4, once again keyed to last name permits any data about any name in the file to be changed. Once again if more than one person with the same last name exists in the file then the duplicate name option will permit any of them to be retrieved.

Options 5-8 are for saving and recalling files from cassette and/or diskette. About the only thing really necessary to remember is not to save data on top of old data unless you desire the old data to be erased. This is true physically on cassettes and on diskettes if an existing name is used as the newly saved mail file name.

Option 9 will print the entire list on a line printer in 1 inch mailing label format. The names will be printed in the order in which they were entered.

Option 10 prints an alphabetical list of the names along with their corrsponding membership numbers on the line printer.

Option 11 permits an individual name to be printed on the line printer in mailing label format. Once again the duplicate names option appears.

MARKUP

Markup is the gross profit a businessman makes on an item expressed either as a per cent of cost or selling price. **MARKUP** has a MENU containing three options. The user may either (1) determine selling price given cost and markup with markup based upon cost or (2) determine selling price given cost and markup with markup based upon selling price or (3) determine markup based upon cost and markup based upon selling price given cost and selling price.

The following example will explain how the program works. Mr. James pays $130 for a Superdiskette. He wishes to sell the item at a markup of 42 %. What should the selling price be if markup is based on cost? What if based on selling price?. In addition, if he sells the item for $180 then what is the markup?

We run the program and after some initial documentation we see:

```
          MENU

1. COMPUTE GROSS MARKUP BASED ON COST
2. COMPUTE GROSS MARKUP BASED ON SELLING PRICE
3. COMPUTE GROSS MARKUP GIVEN COST AND SELLING PRICE

MAKE YOUR CHOICE? <1><ENTER>

GROSS MARKUP BASED UPON COST

ENTER COST IN DOLLARS? <130><ENTER>
ENTER MARKUP AS %    ? <42><ENTER>
SELLING PRICE IS     $184.60

PRESS ENTER TO CONTINUE? <ENTER>
```

We return to the MENU and choose option 2. We see:

```
GROSS MARKUP BASED UPON SELLING PRICE

ENTER COST IN DOLLARS? <130><ENTER>
ENTER MARKUP AS %    ? <42><ENTER>
SELLING PRICE IS $224.14

PRESS ENTER TO CONTINUE? <ENTER> -We return to MENU and choose option 3.

GROSS MARKUP GIVEN COST AND SELLING PRICE
INPUT COST           ? <130><ENTER>
INPUT SELLING PRICE  ? <180><ENTER>

GROSS MARKUP BASED UPON COST IS 38.462%
GROSS MARKUP BASED UPON SELLING PRICE IS 27.778%
```

MERGANAL

In evaluating a potential merger or acquisition the supposition is that the acquiring firm is gaining an income stream and that it will pay more for a greater income stream, i.e. for greater after tax earnings. The acquring company will also pay more if the acquisition adds to the company's credit line. **MERGANAL** attempts to determine how much one company should offer to totally buy out a second company.

Input includes present EBIT (earnings before innterest and taxes), present income tax rate, present market value of the common stock of the company, and present value of the firm's debt. In addition to this we enter the expected future EBIT, future tax rate, and the maximum amount the acquring company could hold as debt on the acquired company. The output is the value of the acquired company. Lets look at the following example:

ABC company want to take over DEF company. DEF now has $1,500,000 in debt. The market value of its stock is $3,000,000. The income tax rate is 48%. In its latest report DEF had EBIT of $800,000. An acquired DEF would have EBIT of $1,000,000 , it is estimated. A total of $2,500,000 could be borrowed on DEF (including the $1.5 million above). The business tax rate will soon decrease to 45%. How much should ABC pay for all the stock of DEF?

To solve the problem we run the program and after some initial documentation we see:

```
PRESENT TOTAL VALUE OF FIRM'S STOCK? <3000000><ENTER>
EARNINGS BEFORE INTEREST AND TAXES ? <800000><ENTER>
INCOME TAX RATE (%)                ? <48><ENTER>
MARKET VALUE OF FIRM'S DEBT        ? <1500000><ENTER>
HOW MUCH DEBT CAN FIRM HANDLE      ? <2500000><ENTER>
WHAT WILL FUTURE TAX RATE BE       ? <45><ENTER>
ENTER FUTURE EARNINGS BEFORE INTEREST AND TAXES ? <1000000><ENTER>

VALUE OF FIRM FOR ACQUISITION = $ 6122600

PRESS ENTER FOR MORE COMPUTATION ?
```

Naturally, in a real merger analysis, more than this simple computation would be performed. The value given by the program is a good ballpark figure for negotiation.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

MORTGAGE/A

MORTGAGE/A is a program which will print a mortgage amortization table. In particular, for any mortgage, if the user inputs the principal,the monthly payment, the term of the mortgage and the annual interest rate this program will print a month by month summary of allocation of the mortage payment to interest and amortization for each year of the mortgage. In this summary is also a month by month record of the balance of the principal yet to be paid. In fact the following is one portion of such a printout:

```
PRIN.=$35000.00  PAYMENT=$244.73 TERM= 30.00 YRS. RATE= 7.5%
 NO.       INTEREST      AMORTIZED    BALANCE

 1         218.75          25.98      34974.02
 2         218.59          26.14      34947.88
 3         218.42          26.31      34921.57
 4         218.26          26.47      34895.10
 5         218.09          26.64      34868.47
 6         217.93          26.80      34841.66
 7         217.76          26.97      34814.69
 8         217.59          27.14      34787.56
 9         217.42          27.31      34760.25
10         217.25          27.48      34732.77
11         217.08          27.65      34705.12
12         216.91          27.82      34677.30
YR=1      2614.06         322.70
```

An identical table is constructed for each year of the mortgage.Thus in the above mortgage there would have been 30 such tables printed. An option to print these tables on a line printer exists.

Of course to get a table such as the above the initial mortgage information must be input into the program. This is done at the beginning of the program with several input statements. The questions and the corresponding answers for the above are:

```
MORTGAGE PRINCIPAL = ?_  We press 35000 <ENTER>
MONTHLY PAYMENT    = ?_  We press 244.73 <ENTER>
TERM OF MORTGAGE   = ?_  We press 30 <ENTER>. Note term is in years.
ANNUAL INT.RATE(%) = ?_  We press 7.5 <ENTER>
```

At the completion of this the program commences to print out the mortgage tables. There are two other questions which one must answer at the beginning of the program. One of these asks if line printer output is desired while the other asks if we wish the display to halt at the end of each year for examination.

If you want to know what each payment on a loan is going for then you might try this program.

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

MULTMON

This very simple program computes the amount of time necessary for an investor to multiply his/her money by any amount when it earns compounded interest. The program gives two major options. The interest rates may be specified either as annual interest rates or as per period interest rates. The per period rate is the actual rate of interest used by the bank in one period interest calculation, e.g. 1.5% every three months. In case the interest rate is quoted annually is quoted it is necessary for the user to specify the number of times money is compunded annually, e.g. quarterly interest crediting involves 4 compoundings per year.

The basic annual formula used is $A=a*(1 + p/NC)^n$ where A is the final amount, a is the initial amount, p is the annual interest rate, NC is the number of interest compoundings per year, and n is the total number of compoundings so far. In case interest is compounded daily NC may be either 360 or 365 (or 366 in leap years). Banks use both divisors. Using 360 gives a higher effective interest rate.

The output of the program really involves a bracket of two numbers. The money will actually double in the time period between these two numbers.

Lets look at the following example where interest rates are quoted annually. User input is between { } below but not in the program.

```
ENTER ANNUAL INTEREST RATE IN %,E.G. 5% ENTERED AS 5? <6><ENTER>

WE NOW WANT THE NUMBER OF COMPOUNDINGS PER YEAR.
IF THIS IS DAILY SOME BANKS USE 360 AND SOME USE 365
ENTER NUMBER? <4><ENTER>

ENTER THE FACTOR BY WHICH YOU DESIRE TO INCREASE YOUR FUNDS.
E.G. TO DOUBLE INPUT 2,TRIPLE 3,ETC. ? <2><ENTER>

INTEREST RATE AND TIME MEASURED ANNUALLY

ANNUAL INTEREST RATE EQUALS 6 PER CENT
INTEREST COMPOUNDED 4 TIMES PER YEAR

TIME NECESSARY TO MULTIPLY ACCOUNT BY 2 EQUALS 11.639 YEARS

ACTUALLY THIS TAKES BETWEEN 46 AND 47 INTEREST PERIODS
OR BETWEEN 11.5 AND 11.75 YEARS
THE EXACT AMOUNT OF TIME NEEDED DEPENDS UPON HOW INTEREST
IS CREDITED WITHIN AN INTEREST PERIOD.

ENTER Y FOR MORE COMPUTATIONS, N TO END ? { N }
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

NCFANAL

When a business invests in aproject there is an initial cash outlay followed, hopefully by a sequence of cash inflows over several years. In general for every year during the useful life of the project there are both cash inflows and cash outflows associated with the project. There is also a depreciation charge. When these are amalgamated there is a net profit each year which is taxed at the prevailing corporate tax rate. After taxes are taken out what is left is the net cash flow. It is the net cash flow which is really important to a firm as this is really the only thing which can be paid out to stockholders as dividends or used for new investments.

NCFANAL prints out a table giving net cash flow in the following type of situation: There is an initial cash outlay for the project, a depreciable cost, a given useful lifetime, a salvage value, and a corporate tax rate which is input. In addition for each year during the project's useful lifetime the user may input the total cash inflows and (pretax) cash outflows. Straight line depreciation is used for the table.

The following example shows how the program operates: There is a project involving an initial cash outlay of $10000 even though $12000 is allowed as the basis for depreciation. The project has a useful life of 4 years and may be salvaged for $4000 at that time. Estimated cash inflows during each of the four years are 3000,4000,4500,and 5000 respectively. Cash outflow is estimated to equal $1000 per year. The corporate tax rate is 48%.

We RUN the propgram and after some initial documentation we see:

```
INPUT CASH COST OF THE PROJECT          ?<10000><ENTER>
INPUT DEPRECIABLE COST OF THE PROJECT   ?<12000><ENTER>
INPUT USEFUL LIFE OF PROJECT IN YEARS   ?<4><ENTER>
SALVAGE VALUE AT END OF USEFUL LIFE     ?<4000><ENTER>
INPUT CORPORATE INCOME TAX RATE(%)      ?<48><ENTER>

CASH FLOWS FOR YEAR 1 NOT INCLUDING INCOME TAXES
     CASH INFLOWS   = ?<3000><ENTER>
     CASH OUTFLOWS  = ?<1000><ENTER>
---------
---------
CASH FLOWS FOR YEAR 4 NOT INCLUDING INCOME TAXES
     CASH INFLOWS  = ?<5000><ENTER>
     CASH OUTFLOWS = ?<1000><ENTER>
```

YR	CASH INFLOW	CASH OUTFLOW	DEPREC-IATION	NET INCOME	TAXES	NET CASH FLOW
0	0	10000	0	0	0	-10000
1	3000	1000	2000	0	0	2000
2	4000	1000	2000	1000	480	2520
3	4500	1000	2000	1500	720	2780
4	5000	1000	2000	2000	960	7040

YEAR 4 CASH FLOW INCLUDES SALVAGE VALUE SALE OF $4000

NPV

When comparing investment projects involving different initial costs, different expected cash inflows and outflows during their lifetimes, and even different lifetimes it is necessary to have some yeardstick for comparison. One commonly used such yardstick is the Net Present Value(NPV) of all cash flows into and out of a project. This idea takes note of the concept that receiving $1 today is better than receiving $1 in 1 year, and that is better than receiving the dollar in two years, etc. In using net present value future cash flows are "discounted" to "equivalent" immediate amounts. These amounts are summed to get a total value today. The rate of interest used in this discounting is called the discount rate or required rate of return and depends upon the project's risk. The basic idea is that projects with positive NPV's may be good investments.

NPV does a net present value computation with the input of the the required rate of return, the initial cash cost of the project, and the total cash flows for each year in the useful life of the project.

The following example shows how the program operates: Jane Brane is thinking of investing in a project. The initial cost is $10,000. Based upon what other similar projects yield Jane feels that a required rate of return of 16% is justified. The project will bring in cash of $3000 during each of the first three years and then $4000 during each of the next two years. After that there are no cash flows which the project will generate. Find the NPV of the cash flows of the project.

We RUN **NPV** and after some initial documentation we see:

```
INPUT FOR NEW PROJECT

INPUT REQUIRED RATE OF RETURN (%) ?<16><ENTER>

INPUT CASH COST OF PROJECT       ?<10000><ENTER>
ENTER CASH FLOW FOR PERIOD 1 (-.01 TO END)?<3000><ENTER>
ENTER CASH FLOW FOR PERIOD 2 (-.01 TO END)?<3000><ENTER>
ENTER CASH FLOW FOR PERIOD 3 (-.01 TO END)?<3000><ENTER>
ENTER CASH FLOW FOR PERIOD 4 (-.01 TO END)?<4000><ENTER>
ENTER CASH FLOW FOR PERIOD 5 (-.01 TO END)?<4000><ENTER>
ENTER CASH FLOW FOR PERIOD 6 (-.01 TO END)?<-.01><ENTER>

NET PRESENT VALUE AT 16 % EQUALS      $851
DO YOU WANT ANOTHE NPV FOR SAME CASH FLOWS BUT CHANGING
INTEREST RATE(Y/N) ?
```

A Y above permits another calculation by changing the interest rate but not touching the cash flows. An N begins the processing of a new project. The positive NPV of $851 means that the project does yield more than 16%.

OPTLOSS

This program assumes that there are several possible states of nature. There are also several actions which may be taken. If the state of nature(S of n) is known then each action leads to a known monetary result. This result is called a conditional profit. For example,if the payoff when action 3 is taken and S of N 2 is true equals $35 then $35 is the conditional profit of action 3 given S of N 2. Each action-S of N pair yields a conditional profit. If we look at the coditional profit of each action with respect to a particular fixed S of N then there is a largest profit. The amount by which each of the other profits corresponding to that S of N differ from this largest profit is called the **opportunity loss** of the corresponding action for the particular S of N. This program constructs a so-called opportunity loss profit table, which is just a rectangular matrix, the elements of which are opportunity losses. We also assume that we know the probability of occurrence of each S of N and compute the expected opportunity loss for each action.

In the following an opportunity loss profit table is constructed in a case with two possible S of N's and 3 possible actions. S of N 1 occurs with probability .3 and S of N 2 with probability .7. When S of N 1 is true then our payoffs are 2 for action 1, 5 for action 2, and 3 for action 3. When S of N 2 is true the payoffs are 5 for action 1,7 for action 2 and 2 for action 3. We RUN the program and see:

```
ENTER NUMBER OF POSSIBLE STATES OF NATURE? <2><ENTER>
ENTER PROBABILITY OF S OF N 1 ? <.3><ENTER>
ENTER PROBABILITY OF S OF N 2 ? <.7><ENTER>

ENTER NUMBER OF POSSIBLE ACTIONS(<=5)? <3><ENTER>
NO PAYOFF MORE THAN 9999.99

ACTION 1
     PAYOFF WHEN S OF N IS 1 ? <2><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <5><ENTER>
ACTION 2
     PAYOFF WHEN S OF N IS 1 ? <5><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <7><ENTER>
ACTION 3
     PAYOFF WHEN S OF N IS 1 ? <3><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <2><ENTER>

    OPPORTUNITY LOSSES OF ACTION GIVEN S OF N

S N  PR                      ACTION
          ACTION 1      ACTION 2      ACTION 3
 1  0.30      3.00          0.00          2.00
 2  0.70      2.00          0.00          5.00

EXP VAL       2.30          0.00          4.10
```

OPTWRITE

This program computes some useful totals for potential option writers. The user of this program will generally be a ratio writer and will simultaneously sell a fixed number of options as well as buy or short a given number of shares of the underlying stock. With this program he will be able to compute the maximum dollar profit he may realize, the dollar investment required for the deal, the maximum return on investment, and the upper and lower breakeven points of stock price for the deal.

Program input icludes whether puts or calls are being considered, # of option contracts sold, price per option, total commission paid on options, exercise price, # shares bought or shorted, price per share of stock, average commission per stock share as % of stock share price, dividends per share paid by stock during options holding period, maintenance requirement.

Program output includes maximum profit, initial investment, maximum return, upper break-even point, lower break-even point.

The following example shows how the program operates: Jim Dollar has the following writing possibility. He wishes to write 5 calls at an exercise price of 30 on a stock currently selling for $29 per share. As collateral he will buy 300 shares of the stock. Stock commissions run about .5% of value while the commission on the 5 options will be $25. The stock is expecte to pay a commission of $1.25 per share before the option expires. His broker will require Jim to put up an additional $1000 as maintenance to secure this position. Current price of the calls equals $4 per share.

We RUN the program and after some initial documentation we see:

```
INPUT C IF OPTION IS CALL, P IF PUT     ? <C><ENTER>
ENTER NUMBER OF OPTION CONTRACTS SOLD           ? <5><ENTER>
ENTER PRICE PER OPTION                          ? <4><ENTER>
ENTER TOTAL COMMISSION PAID ON OPTIONS          ? <25><ENTER>
EXERCISE PRICE                                  ? <30><ENTER>
# OF SHARES BOUGHT OR SHORTED                   ? <300><ENTER>
PRICE PER SHARE OF STOCK                        ? <29><ENTER>
AV.COMM. ON ST.SHARE AS % OF SHARE PRICE        ? <.5><ENTER>
DIVIDENDS PER SHARE PAID BY STOCK               ? <1.25><ENTER>
MAINTENANCE REQUIREMENT                         ? <1000><ENTER>


MAXIMUM PROFIT                  =    $2561.50
INITIAL INVESTMENT              =    $7768.50
MAXIMUM RETURN                  =       32.97 PER CENT
UPPER BREAK EVEN POINT          =       42.45 PER CENT
LOWER BREAK EVEN POINT          =       21.42 PER CENT
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

PAYNET

Very often a new employee is very disappointed upon getting his first envelope. It is a light lighter than expected due to an onerous tax burden. This employee asked for too little in his new place of employment. Most people know about how much net pay they need in a fixed amount of time. Most salaries are described as gross salaries. This program attempts to bridge the gap between the two extremes.

The person using this program picks a desired net pay amount per pay period. He then enters fixed effective federal, state, local, and FICA withholding tax rates. There is an option for a fixed deduction which may occur every pay period. This might be union dues, hospitalization, etc. The program then computes the gross pay needed to attain the desired net. Armed with this gross amount the employee knows what to ask for from his present employer or in the job market.

Lets look at the following example: John needs net pay of $400 every pay period. At this pay level the effective federal withholding rate is about 23%., that of the state is 7%, that of the city is 1%. FICA withholding equals 6.85%. There is also a $21 cut which goes to the union. What gross must John make?

We RUN **PAYNET** and after some initial documentation we see:

```
ENTER NET PAY AMOUNT DESIRED    ? <400><ENTER>
ENTER FEDERAL WITHHOLDING TAX % ? <23><ENTER>
ENTER STATE WITHHOLDING TAX %   ? <7><ENTER>
ENTER LOCAL WITHHOLDING TAX %   ? <1><ENTER>
ENTER FICA WITHHOLDING  %       ? <6.85><ENTER>
ENTER DOLLAR AMOUNT FOR ANY FIXED WITHHOLDING? <21><ENTER>

GROSS PAY NEEDED IS $677.39

PRESS ENTER TO CONTINUE?
```

Note that if this is weekly pay the program tellus that to take home $20,800 the worker would have to earn over $35,000 gross.

PAYROLL2

PAYROLL2 is a brief in memory payroll system. It is designed for those who do not desire to look up amounts of Federal and FICA withholding and keep records of such on hard copy ledger sheets. The program is set up for weekly pay periods and is not set up for any state or local taxes, though these may be handled through the "OTHER DEDUCTIONS?" question.

The program will set up a data file which will store yearly payroll data on up to 25(on 16k machines) employees. This data file, after being initially set up will hold 25 employee's data although some of these employees will be "blank" e,ployees not appearing in any printout. The format of the file is something like that used for the customer file in ACCTREC and ACCTPAY.

The MENU shows what the program is capable of:

MAIN MENU

1.MASTER FILE MAINTENANCE
2.READ FILE IN FROM TAPE
3.WRITE FILE OUT ONTO TAPE
4.READ FILE IN FROM DISK
5.WRITE FILE OUT ONTO DISK
6.COMPUTE PAYROLL
7.POST PAYROLL TO YTD TOTALS
8.REPORT GENERATOR
9.END OF PERIOD FILE MAINTENANCE

On first getting the system up it is necessary to use option 1 to do MASTER FILE MAINTENANCE. The employees have numbers from 1 to 25. For each employee the following data is kept on file:

EMPLOYEE NAME
SOCIAL SECURITY NUMBER
ADDRESS 1ST LINE
ADDRESS 2ND LINE
PAY RATE
NUMBER OF EXEMPTIONS
MARRIED OR SINGLE
YEAR TO DATE GROSS EARNINGS,FEDERAL WITHHOLDING,FICA WITH.
YEAR TO DATE SPECIAL WITHHOLDING
MONTH TO DATE OF THE ABOVE
CURRENT OF ALL OF THE ABOVE, AND A DATE

After the initial file maintenance in which employee data is entered the master file should be written onto cassette or diskette. There is no automatic saving of updated data. This must be done by the user using options 3 or 5 from the MAIN MENU.

In order to compute payroll option 6 is used. A pay date will be entered ands then the user will see:

ENTER ID #(0 TO END),REG HRS,PREMIUM HRS?<3,40,5><ENTER>

We have just said that employee 3 has worked 40 hours at regular pay and 5 hours at overtime(time and 1/2)

ENTER SPECIAL WITHHOLDING ? <5><ENTER>

Employee 3 has an extra $5 withheld.

After this entry the program will compute employee 3's gross and net wages along with proper deductions. These amount are put in the current fields of the master. All tables are weekly in this version. YTD and Quarter to date totals have not been changed.

All employees are entered as in the above. If any entry is bad that employee may be recomputed as no permanent records have been made. During this time a Payroll register may be printed by using the REPORT GENERATOR.

When the data for the pay period is correct, option 7 may be used. This updates the YTD and Quarterly totals. After this the only way to correct a bad entry is to do MASTER FILE MAINTENANCE.

The report generator will print up quarterly and year to date totals, in addition to giving the necessary data for W-2 forms.

This system is a nice system that beats handling this data manually. The only real weakness is the fact that state and local withholding are not directly supported.

PORTVAL9

PORTVAL9 is a simple list program which is set up to keep track of a investor's portfolio of stocks, though it can also be used for bonds and with proper interpretation for income producing property. What the program does is store the following information for each security held:

Stock ticker code
Company name
Date of purchase
units purchased
purchase price per unit
present price per unit
yearly dividends per unit

The data may be stored on cassette or diskette and may be recalled at any time. When a security is sold it may be removed from the list. A profit analysis of paper profits may also be performed. The system does not presently consider securities which are sold. However, by making the selling price per unit equal to present price per unit and then not changing this we can find out how much profit was made on the position.

When we begin to run the program the following MENU appears:

```
              MAIN MENU

    1. ADD TO FILE OR BEGIN NEW FILE
    2. READ FILE IN FROM TAPE
    3. WRITE FILE OUT ONTO TAPE
    4. READ FILE IN FROM DISK
    5. WRITE FILE OUT ONTO DISK
    6. EDIT FILE
    7. LIST FILE AND PROFIT ANALYSIS
    8. ROUTINE ACTUALLY DELETING RECORD FROM FILE

ENTER YOUR CHOICE?_
```

Operation should be straightforward. The only thing that is important to remember is that we are working with a sequential file when stored on cassette or diskette. This sequential file must be explicitly read into memory before any operations may commence(options 2 or 4) and must be explicitly written onto cassette or diskette at the conclusion of any operations which have the effect of adding to or editing the file. The only exception is when the file is first begun. In that case option 1 is used right away. However, at the conclusion of input the data should be saved onto cassette or diskette. Then, if there is subsequent machine or operator error the file will still be saved.

There is option for line printer output.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

PRINDLAS

The buying power of the dollar varies over time. For example to compare the relative price of a hot dog in 1956 to the price of a hot dog today we must determine the buying power of a dollar today compared to that of 1956. This is done with index numbers. An index number, generally written down in percentage form, compares the prices in a base period(corresponding to 1956 above) to the prices in a reference period(corresponding to now in the above). An index number greater than 100 means prices have gone up, less than 100 means prices have gone down. To be more precise, an index number of 135 e.g. means prices have increased by 35% from the base period to the reference period. An index number of 90 means prices have decreased by 10% during that time interval.

It is not so simple to determine relative price levels since between two periods of time new items come on to the market and the relative use of other items changes. The Laspeyres price index which is computed in this program is used by U.S. Department of Labor to compare price levels. Several items which were sold in both the base year and the reference year are chosen. The price of each item is determine for each of the periods. These prices are weighted with the quantity of the item sold in the base year. Thus two weighted averages are determined. The ratio of these multiplied by 100 give the Laspeyres index.

Lets see how this works with the following three items:

ITEM	Quanity(base)	Price/unit(base)	Price/unit(ref.)
bread	37	$.25	$.57
milk	10	$.32	$1.00
meat	3	$1.21	$2.00

We run the program and after some initial documentation we see:

```
FOR ITEM # 1 ENTER THE FOLLOWING:
          BASE YEAR QUANTITY(NEGATIVE ENDS)? <37><ENTER>
          BASE YEAR PRICE                  ? <.25><ENTER>
          CURRENT YEAR PRICE               ? <.57><ENTER>
FOR ITEM # 2 ENTER THE FOLLOWING:
          BASE YEAR QUANTITY(NEGATIVE ENDS)? <10><ENTER>
          BASE YEAR PRICE                  ? <.32><ENTER>
          CURRENT YEAR PRICE               ? <1.00><ENTER>
FOR ITEM # 3 ENTER THE FOLLOWING:
          BASE YEAR QUANTITY(NEGATIVE ENDS)? <3><ENTER>
          BASE YEAR PRICE                  ? <1.21><ENTER>
          CURRENT YEAR PRICE               ? <2.00><ENTER>
FOR ITEM # 4 ENTER THE FOLLOWING:
          BASE YEAR QUANTITY(NEGATIVE ENDS)? <-1><ENTER>

THE LASPEYRES PRICE INDEX EQUALS  230.659
```

PRINDPA

The buying power of the dollar varies over time. For example to compare the relative price of a hot dog in 1956 to the price of a hot dog today we must determine the buying power of a dollar today compared to that of 1956. This is done with index numbers. An index number, generally written down in percentage form, compares the prices in a base period(corresponding to 1956 above) to the prices in a reference period(corresponding to now in the above). An index number greater than 100 means prices have gone up, less than 100 means prices have gone down. To be more precise, an index number of 135 e.g. means prices have increased by 35% from the base period to the reference period. An index number of 90 means prices have decreased by 10% during that time interval.

It is not so simple to determine relative price levels since between two periods of time new items come on to the market and the relative use of other items changes. The Paasche price index is computed in this program. It is computed in the following way. Several items which were sold in both the base year and the reference year are chosen. The price of each item is determine for each of the periods. These prices are weighted with the quantity of the item sold in the reference year. Thus two weighted averages are determined. The ratio of these multiplied by 100 give the Paasche index.

Lets see how this works with the following three items:

ITEM	Quanity(ref)	Price/unit(base)	Price/unit(ref.)
bread	40	$.25	$.57
milk	15	$.32	$1.00
meat	3	$1.21	$2.00

We run the program and after some initial documentation we see:

```
FOR ITEM # 1 ENTER THE FOLLOWING:
          CURRENT YEAR QUANTITY(NEGATIVE ENDS)? <40><ENTER>
          BASE YEAR PRICE                     ? <.25><ENTER>
          CURRENT YEAR PRICE                  ? <.57><ENTER>
FOR ITEM # 2 ENTER THE FOLLOWING:
          CURRENT YEAR QUANTITY(NEGATIVE ENDS)? <15><ENTER>
          BASE YEAR PRICE                     ? <.32><ENTER>
          CURRENT YEAR PRICE                  ? <1.00><ENTER>
FOR ITEM # 3 ENTER THE FOLLOWING:
          CURRENT YEAR QUANTITY(NEGATIVE ENDS)? <3><ENTER>
          BASE YEAR PRICE                     ? <1.21><ENTER>
          CURRENT YEAR PRICE                  ? <2.00><ENTER>
FOR ITEM # 4 ENTER THE FOLLOWING:
          CURRENT YEAR QUANTITY(NEGATIVE ENDS)? <-1><ENTER>

THE PAASCHE PRICE INDEX EQUALS  237.656
```

PROFIND

The Profitability Index of an investment project, sometimes known as the benefit-cost ratio, is simply the net present value of the future net cash flows divided by the initial cash outlay. The net present value is computed using an interest rate equal to the rate which projects with about the same risk as the one in question are then currently yielding. This is sometimes called the required rate of return and if information about about it is not available then the investor sometimes just estimates what return he would like to earn from the project and use this rate. The general rule when using profitability index as a measure of the acceptability of an investment is that the investment is acceptable if the profitability index is greater than 1 and not acceptable if the profitability index is less than 1.

The criterion of profitability index does not always give the same set of investment projects as that of present value when there is a limited amount of cash available for the investor to invest. However, as a general rule, the larger the profitability index the more bang the investor gets for his bucks.

To run **PROFIND** the user must input the desired rate of return for the project, the initial cash outlay, and the cash flow for each year during which the project affects cash flows. A cash outflow after the initial cost of the project is entered as a negative value.

The following example shows how the program runs. Mr. Jones is thinking about an investment project. After an initial outlay of $18,000 he expects to receive 5600 per year for five years. His desired (required) rate of return on the project is 10%. Compute the profitability index:

We run the program and after some initial documentation we see:

```
ENTER RETURN DESIRED FOR THIS PROJECT(%) ? <10><ENTER>
ENTER INITIAL CASH OUTLAY FOR THIS PROJECT ? <18000><ENTER>
ENTER CASH FLOW FOR YEAR 1 (-9999 ENDS) ? <5600><ENTER>
ENTER CASH FLOW FOR YEAR 2 (-9999 ENDS) ? <5600><ENTER>
ENTER CASH FLOW FOR YEAR 3 (-9999 ENDS) ? <5600><ENTER>
ENTER CASH FLOW FOR YEAR 4 (-9999 ENDS) ? <5600><ENTER>
ENTER CASH FLOW FOR YEAR 5 (-9999 ENDS) ? <5600><ENTER>
ENTER CASH FLOW FOR YEAR 6 (-9999 ENDS) ? <-9999><ENTER>

INITIAL CASH COST OF PROJECT =  18000
PRESENT VALUE OF FUTURE CASH FLOWS = 21228.4
PROFITABILITY INDEX EQUALS 1.17936

PRESS ENTER FOR MORE COMPUTATION?
```

Naturally the -9999 is not considered in the computation.

PVAL

If you were asked the question of whether you would prefer $1000 now or $1000 five years from now you would probably say you would prefer to have the money now. There are several reasons for this. First of all, we generally prefer our gratifications now rather than in the future. Secondly, in a time of inflation we can buy more with $5000 now than with $5000 in five years. A third reason is very interesting. If we got $5000 now and put the money in the bank earning interest then at the end of five years we would have considerably more than $5000. This argument ignores the effects of taxes but if we assume that tax rates will be about the same in five years as they are now then our arguments still hold, only we will be talking about less than $5000 , i.e. the initial amount after taxes.

The conclusion we should come to from the above paragraph is that $5000 now is worth at least $5000 + the interest it will earn five years hence. The problem which we come across in **PVAL** is the following: Suppose we have a guarantee of a certain amount of money, say $5000, at some fixed time in the future. How much is that future $5000 worth now ? Essentially, what the last question is asking is what amount of money in your pocket now would be worth the same amount to you as $5000 at that fixed time in the future. For example you would probably rather have $5000 in three years than $1 now. However, you would probably rather have $4500 now rather than $5000 in three years. Somewhere between $1 and $4500 there is an amount where you would not be quite sure wheteher you would prefer the $5000 or three years or that amount now. We can call this the present value of $5000 in three years.

If we know that the interest rates are going to be very high during the next couple of years then present values of future amounts are going to go down as less money in the bank now will still be able to earn significant interest in years to come.

In the present program we have a future AMOUNT TO BE OBTAINED which we input. We also input an interest rate (per compounding period) with a special option for rates which are quoted annually, and the number of periods into the future this amount will be obtained. The program then computes the present value of the future amount. It is not easy to estimate the interest rate as this is a future amount.

As an example what is the present value of $5000 in 4 years if an interest rate of 2% per quarter is used and interest is compounded quarterly? We run the program and see:

```
ARE INTEREST RATES QUOTED ANNUALLY(Y/N)?_ (We answer N <ENTER>)
INTEREST RATE,I,PER PERIOD?_  We enter 2 <ENTER>.
AMOUNT TO BE OBTAINED ? - We enter 5000 <ENTER>
NUMBER OF PERIODS     ?_  We enter 16 <ENTER>

PRESENT VALUE = 3642.23
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

QUEUE1

A QUEUE is a waiting line. There are queues in almost all business systems ranging from supermarket checkout lines to lines at airport runways. A SERVER provides service to the items or customers on the queue. An example of a server is a runway at the airport. If there is a system involving queues then the size of the resulting waiting lines are affected by such variables as how quickly customers arrive into the system desiring service (mean arrival rate), how long it takes to service the customer once he is off the waiting line (mean service rate) and the number of servers. In general, the basic results bring no surprises. More servers bring about shorter waiting lines. Greater service rates relative to arrival rates also bring about shorter lines.

In **QUEUE1** we analyze a single server queue and calculate some important parameters relative to waiting line length and time that a customer is in the system. Our assumptions are as follows:

The number of arrivals per unit time is random but the mean arrival rate is known.(The theoretical name for arrivals is Poisson distribution.
Service times are random but mean service rate is known (theoretical name for for distribution of service time is exponential distribution)
Queue is first come - first served.
Mean arrival rate is less than mean service rate.
Population of customers is very large.

Lets run an example where the mean arrival rate is 4 units per hour while the mean service rate is 6 units per hour.
We RUN the program and after some initial documentation we see:

```
THE FOLLOWING ARE IN NUMBER PER UNIT TIME

     ENTER MEAN ARRIVAL RATE OF UNITS? <4><ENTER>
     ENTER MEAN SERVICE RATE OF UNITS? <6><ENTER>

AVERAGE LENGTH THE OF QUEUE          = 1.33333
AVERAGE LENGTH OF THE SYSTEM         = 2
AVERAGE WAITING TIME ON QUEUE        = .333333
AVERAGE TIME WAITING IN SYSTEM       = .5
PROBABILITY SYSTEM IS BUSY           = .666667

SYSTEM TIME IS WAITING TIME PLUS SERVICE TIME.

PRESS ENTER FOR MORE?
```

QUEUECB

A QUEUE is a waiting line. There are queues in almost all business systems ranging from supermarket checkout lines to lines at airport runways. A SERVER provides service to the items or customers on the queue. An example of a server is a runway at the airport. If there is a system involving queues then the size of the resulting waiting lines are affected by such variables as how quickly customers arrive into the system desiring service (mean arrival rate), how long it takes to service the customer once he is off the waiting line (mean service rate) and the number of servers. In general, the basic results bring no surprises. More servers bring about shorter waiting lines. Greater service rates relative to arrival rates also bring about shorter lines.

A customer waiting on line represents an opportunity loss to the serving institution. In this program we quantify this number as a cost per customer on line per unit time. We also quantify the cost of a server on a cost per unit time basis. Then with fixed service and arrival rates this program will, for various numbers of possible servers compute the total cost of the queuing system as the sum of the cost of the servers per unit time (cost = number of servers * cost of 1 server/unit time) and the cost of the waiting line/unit time (cost = average # of customers on waiting line * cost of 1 customer waiting/unit time).

We try the following example: On average 12 cars/minute arrive at the toll gate of the QUADBOROUGH BRIDGE. A single tolltaker can serve 6 cars per minute. A tolltaker makes $6 per hour($.10/minute) while it is estimated that it cost a motorist $.17/minute to wait on line. How many tolltakers should there be?

We RUN the program and after some initial documentation we see:

```
ENTER MEAN SERVICE RATE(ITEMS PER UNIT TIME)  ? <6><ENTER>
ENTER MEAN ARRIVAL RATE(ITEMS PER UNIT TIME)  ? <12><ENTER>
ENTER WAITING LINE COST($ PER ITEM ON LINE
          PER UNIT TIME                       ?<.17><ENTER>
ENTER COST OF A SERVER($ PER UNIT TIME)       ?<.10><ENTER>
```

SERVERS	AVG.QUEUE LENGTH	SERVER COST/ UNIT TIME	QUEUE COST/ UNIT TIME	TOTAL COST/ UNIT TIME
3	0.89	0.30	0.15	0.45
4	0.16	0.40	0.03	0.43
5	0.04	0.50	0.01	0.51
6	0.01	0.60	0.00	0.60
7	0.00	0.70	0.00	0.70
8	0.00	0.80	0.00	0.80

Note how total cost is minimized in the 4 server situation.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

REGWITH

Suppose you have worked hard all your life and are now ready to retire. In addition to social security and a pension you have save quite a bit and have placed all of these savings into a bank account which earns interest at a fixed rate compounded monthly. You wish to draw a monthly amount from this bank account to supplement your social security and pension payments. You decide that you would like the money in the bank account to last 20 years. If the same amount is withdrawn every month and if the remainder keeps earning interest then how much can be withdrawn every month so that the account is totally depleted in exactly 20 years?

REGWITH is the program the above prospective retiree should use. REGWITH computes the uniform amount which can be withdrawn from an initial investment M times per year for exactly N years with a given nominal interest rate so that nothing is left after the N years. Part of the assumption is that the money remaining is compounded monthly. As will be seen in the following example the amount which may be withdrawn depends heavily upon the interest rate.

Lets look at the following example: Jane has a $12,000 inheritance which according to her uncle's will must be used for housing during her college years. Jane will spend 4 years or 48 months in college and is ready to sign a 4 year lease. The $12,000 is invested in an account earning 11% interest per year, compounded monthly. How much can Jane afford for a monthly lease? Since she must use the money for housing she might as well allow the $12,000 to diminish to 0 by the 48th payment so she runs the REGWITH program. After some initial documentation she sees:

```
AMOUNT OF INITIAL INVESTMENT?_ She presses 12000 <ENTER>
NUMBER OF WITHDRAWALS PER YEAR?_ She presses 12 <ENTER>
NUMBER OF YEARS OF WITHDRAWALS?_ She presses 4 <ENTER>
NOMINAL INTEREST RATE(%) ?_  She presses 11 <ENTER>

REGULAR WITHDRAWAL PERMITTED =   $310.15

PRESS 0 TO END, 1 TO START OVER,2 FOR NEW INTEREST RATE?_
      She wishes to see what would happen at a 5% interest rate so
      presses 2 <ENTER>

NOMINAL INTEREST RATE(%) ?_ She presses 5 <ENTER>

REGULAR WITHDRAWAL PERMITTED =  $276.36
```

Note the difference due to the change in interest rates. A computation like this should give some food for thought for those people who keep their money in non interest bearing accounts.

RENTPRCH

The prime requirement of purchasing rental property is that a profit be available either immediately or sometime into the future. A bank writing a mortgage on the property might want some assurance that the mortgagee has the ability to repay the mortgage. One thing which both the investor and the bank might look for is that total rental income is at least as great as total expenses. These expenses include mortgage payments, tax payments, maintenance costs, insurance costs and other miscellaneous charges which might be lumped with maintenance, if desired.

In **RENTPRCH** the prospective purchaser knows the following information: monthly maintenance costs,monthly insurance costs, monthly tax payments, monthly miscellaneous charges, total monthly rental income and the annual interest rate on mortgages presently being written on rental properties. What the program returns is the size of the mortgage which will have a payment making total expenses just equal to total rental income. This mortgage size may be considered as the value of the property at least from the bank's conservative point of view.

The following is a sample run of the program: We assume that annual mortgage rates are 13.5% for 25 year mortgages on this type of property. A building has rental income of $895 per month. Monthly maintenance has been running at $225 per month, insurance at $30 per month, taxes at $50 per month, and miscellaneous charges at $10 per month. What size mortgage will the bank give for the building if total expenses are to be know more than total rental income?

We RUN the program and after some initial documentation we see:

```
ENTER TOTAL MONTHLY RENTAL INCOME ? <895><ENTER>
ENTER MONTHLY INSURANCE           ? <30><ENTER>
ENTER MONTHLY MAINTENANCE         ? <225><ENTER>
ENTER MONTHLY TAXES               ? <50><ENTER>
ENTER MONTHLY MISCELLANEOUS COSTS ? <10><ENTER>
ENTER ANNUAL MORTGAGE INT RATE(%) ? <13.5><ENTER>
ENTER LENGTH OF MORTGAGE(YEARS)   ? <25><ENTER>

MONTHLY INTEREST = 1.125%
MORTGAGE PAYMENT WILL EQUAL   $580.00

AMOUNT PROPERTY IS WORTH = $49758

PRESS ENTER FOR MORE COMPUTATION?
```

It is interesting what effect lowering interest rates have on the value of the property. In the above if interest rates are lowered to 10%, a 26% decrease we see that the value of the property becomes 62827, a 28% increase.

RRCONST

The rate of return one receives on an investment is a type of average interest rate. Investments may involve different initial outlays and may return different amounts at different time intervals. The internal rate of return (IRR) which is computed in this program is a yardstick by which different investment opportunities may be compared. All other things being equal a higher IRR is better than a lower one. The present program calculates the IRR in a very restricted but common case. The type of problem analyzed here is the following: A given amount is invested now. This is $A. In return for this $B is guaranteed at the end of each of the next N time periods. What is the IRR? The reason this is important is that bank loans are of this type. In terms of method of computation, this program uses a searching routine.

Lets try an example: Mrs. S. has a toothache and her dentist gives Mr.S. a headache with a bill for $2000. Mr. S. borrows $2214 from the bank which he pays back in 36 equal installments of $75. What interest rate did Mr.S. pay on the loan:

To solve this problem we run the RRCONST program and after some initial documentation we see:

INITIAL INVESTMENT ?_ We press 2214 <ENTER>

PERIODIC RETURN = ?_ We press 75 <ENTER>

NUMBER OF PERIODS ?_ We press 36 <ENTER>

The program then does some computations the results of which may be followed in the window in the upper right hand portion of the screen. At completion we see:

RATE OF RETURN IS 1.114% PER PERIOD
IF THIS DATA IS MONTHLY WE MAY MULTIPLY BY 12 TO GET AN
ANNUAL RATE. PRESS <ENTER> FOR THIS COMPUTATION,1 <ENTER>
OTHERWISE? _

We press <ENTER> and see:

ANNUAL RATE OF RETURN IS 13.368%

PRESS <ENTER> FOR MORE COMPUTATIONS?_

Actually we computed the bank's rate of return on the loan. But what the bank gets Mr. S. pays.

The procedure used should work on most bank loans but there is a chance that the searching will not succeed.

RRCONVED

A convertible bond is a bond which, at the holder's option, may be converted into a perdetermined number of shares of stock of the issuing company. Convertible bonds combine the conservative nature of bonds with the more speculative nature of common stock. As such a convertible bond should be worth more than an ordinary bond of the same company and also be worth more than the number of shares of stock that the bond may be turned in for.

The present program computes rate of return on convertible bond purchases. The user inputs the following information:

purchase price of bond
bond coupon payment per year
present value of share of stock
expected annual % increase in value of share of stock
conversion ratio(# of shares bond is convertible into)
desired value of bond when sold

The last variable is an price the holder would like the bond to reach before seling. Given the above information the program computes the expected number of years until the selling price is reached and the rate of return on the bond until that time. An assumption in the program is that the bond will not expire or be called before the desired value is reached.

As an example suppose XYZ Company sells a bond convertible at the holder's option into 50 shares of XYZ common stock. The bond, with a $1000 face value is an 8% bond. Peter buys one of these bonds at par (i.e. $1000). XYZ common is now selling for $18 per share. Peter expects XYZ common to increase in value at a rate of 15% per year. He would like to sell the bond when it reaches a value of $1200. How many years must he wait and what will his return be?

We RUN **RRCONVED** and after some initial documentation we see:

```
ENTER PURCHASE PRICE OF THE BOND          ? <1000><ENTER>
ENTER PRESENT PRICE OF SHARE OF STOCK     ? <18><ENTER>
ENTER CONVERSIO RATIO                     ? <50><ENTER>
ENTER CONVERSION VALUE OF BOND WHEN SOLD? <1200><ENTER>
ENTTER PROJECTED % INCREASE IN STOCK VALUE YEARLY? <15><ENTER>
COUPON PAYMENT PER YEAR                   ? <80><ENTER>

RATE OF RETURN EQUALS     17.2078 % YEARLY
CONVERSION AFTER          2 YEARS

PRESS ENTER FOR MORE ENTRIES?
```

RRVARIN

Suppose an investor invests some funds in a certain proposition. Typically, he puts some money in at the beginning and expects to get something back later. Sometimes he may put money into the proposition several different times. The money coming back to him may come back in one lump sum at the end of a given period of time, it may come back in equal payments at fixed intervals, it may come back in unequal payments at fixed intervals, or it may come back in unequal payments at unequal intervals,etc. There are several ways the investor can compare different propositions with which he may become involved. The method most used involves rate of return computations. In particular we calculate what is called the Internal Rate of Return(IRR) of the investment. IRR is an average rate of return received on an investment. To compute it all the various cash flows are discounted to some fixed point in time,in the case of our program the present.

In our program we have the following situation: The investor invests A$ now and will receive a return of A(I) dollars at the end of year I. I may vary from 1 to 100 with the given dimensioning but this may easily be increased. A(I) may be positive, negative, or zero for any particular I. Thus this program handles variable size inflows and/or outflows. There is nothing which mandayes that the A(I)'s must be different so this program will also handle the case of constant inflows but in this case the computation will be slower than when a program dedicated to constant inflows is used. This is because there is a formula in the constant inflow case. In this program, however, the computer keeps searching for the correct rate of return, getting closer and closer until desired accuracy is reached.

As an example let's find the IRR on the following proposition: The investor invests $20,000 now and expects to get back $5,000 at the end of each of the next 2 years and $6,000 at the end of years 3,4 and 5. At the end of year 5 the investment becomes worthless.To find IRR we run the program and see after some initial documentation:

```
INITIAL INVESTMENT?_  We press 20000 <ENTER>
RETURN YEAR  1 (99999 TO END) ?_ We press 5000 <ENTER>
RETURN YEAR  2 (99999 TO END) ?_ We press 5000 <ENTER>
RETURN YEAR  3 (99999 TO END) ?_ We press 6000 <ENTER>
RETURN YEAR  4 (99999 TO END) ?_ We press 6000 <ENTER>
RETURN YEAR  5 (99999 TO END) ?_ We press 6000 <ENTER>
RETURN YEAR  6 (99999 TO END) ?_ We press 99999 <ENTER>
```

After a few seconds we see:

```
RATE OF RETURN IS 11.88%
```

If you are not experienced in this type of computation beware. If some cash flows are negative wierd things may happen. When comparing different investments, in general those with higher IRR's are preferred.

RTVAL

The privilege of a stockholder to subscribe at a stated price (subscription price) to new issues of common stock in proportion to his present holdings is called a right, a stock right, or a subscription right. This privilege is generally offered for a very limited period, most commonly 30 days. During this period the stockholder may exercise the right (i.e. buy the additional shares at the subscription price), sell the right to some other investor, or do nothing. When the rights offering is announced the Board of Directors establishes a cutoff date. Anyone who purchases the stock before this cutoff date receives the rights for the new issue. At this time the stock is said to be selling Rights-On. After the cutoff date new purchasers of the stock do not get the right and the stock is said to be traded Ex-Rights.

The present program determines various values for rights and for the underlying stock. To do these computations the following information is needed: the market value of 1 share of stock selling rights-on, the subscription proce per share, and the number of rights needed to purchase one share. With this information the program computes the market value of 1 right when the stock is selling rights-on, the theoretical value of one share when it goes ex-rights and the theoretical value of one right when the stock is selling ex-rights.

The following example will show how the program works: SHE Corporation has a Board of Directors meeting and a stock subscription plan is approved. Each present shareholder will receive 1 right for each 3 shares he owns. Three rights and $30 may be traded in by the holder for one share of SHE stock. Right after the meeting SHE stock sells for $56 per share. Determien the value of the rights.

We RUN the program and after some initial documentation we see:

```
ENTER MARKET VALUE OF 1 SHARE RIGHTS-ON? <56><ENTER>
ENTER SUBSCRIPTION PRICE PER SHARE     ? <30><ENTER>
ENTER # RIGHTS NEEDED TO BUY 1 SHARE   ? <3><ENTER>

MARKET VALUE OF 1 RIGHT - RIGHTS-ON          6.5
THEORETICAL VALUE OF 1 SHARE EX-RIGHTS       49.5
THEORETICAL VALUE OF 1 RIGHT EX-RIGHTS       6.5

PRESS ENTER FOR MORE COMPUTATION?
```

RULE78

When a consumer gets a loan at the bank, if he reads the fine print, he will find that if the loan is paid off early interest will be rebated according to the Rule of the 78's. Most bankers do not even know what the rule means. Essentially, it is a way of charging off more interest in early months of the loan when the borrower has most of the bank's money and less in later months when the loan is almost paid off.

RULE78 will print out a table of interest charges for any set of months of a loan if interest is charged off according to the Rule of the 78's. Necessary user input includes the length of the loan, the total interest charge for the loan (written on the loan agreement) , and the months for which we desire the table. As an example suppose the interest on a 48 month loan is $1460 and we wish to know the interest charged from the 24th to 36th month of the loan. We RUN the program and after some initial documentation we see:

```
IS LINE PRINTER OUTPUT DESIRED(Y/N)? <Y><ENTER>
HOW MANY MONTHS(PAYMENTS) IS THE LOAN FOR ? <48><ENTER>
WHAT IS TOTAL INTEREST FOR TERM OF LOAN   ? <1460><ENTER>
WHICH IS 1ST MONTH FOR WHICH OUR INTEREST
      COMPUTATION IS DESIRED              ? <24><ENTER> .
WHICH IS THE LAST MONTH DESIRED           ? <36><ENTER>


TOTAL INTEREST TO BE PAID = $1460
PAYMENT #         INTEREST       INTEREST LEFT
   24               31.04           372.45
   25               29.80           342.65
   26               28.55           314.10
   27               27.31           286.79
   28               26.07           260.71
   29               24.83           235.88
   30               23.59           212.30
   31               22.35           189.95
   32               21.11           168.84
   33               19.86           148.98
   34               18.62           130.36
   35               17.38           112.98
   36               16.14            96.84
```

SALELEAS

SALELEAS is a program which does some computations regarding sale - leaseback analysis. The situation is the following: The owner of a piece of property needs some immediate cash but also needs the use of the property. He is considering two options. He may get a mortgage on the property or he may sell the property and then lease it back for a given number of years from the purchaser. We'll assume that the investor gets the same amount of cash from either alternative and that the terms of the mortgage and the lease would be the same. The interest rate on the lease generally would be different from that of the mortgage. There are different tax ramifications in these two situations and this program helps analyze the differences.

We shall use the following example: HECOMPTRON needs $800000 cash which it can get from a 30 year mortgage on some property it owns. The mortgage rate would be 12.25%. Over the 30 years there will be $600000 depreciation charges on the buildings on the property. It may also enter into a sale-leaseback situation at an interest rate of 13.00 %. The corporate tax rate is 48%. We run the program and after some initial documentation we see:

```
ENTER PRESENT VALUE OF PROPERTY(SALE OR MORTGAGE AMOUNT)? <800000><ENTER>
ENTER THE TERM OF LEASE OR MORTGAGE IN YEARS          ? <30><ENTER>
ENTER ANNUAL INTEREST RATE FOR LEASE IN %             ? <13><ENTER>
ENTER ANNUAL INTEREST RATE FOR MORTGAGE IN %          ? <12.25><ENTER>
ENTER CORPORATE TAX RATE IN %                         ? <48><ENTER>
ENTER TOTAL DEPRECIATION OVER TERM OF LEASE/MORTGAGE  ? <600000><ENTER>


PRESENT VALUE OF PROPERTY                        800000.00
TERM OF LEASE OR MORTGAGE(YEARS)                    30.00
ANNUAL INTEREST RATE FOR LEASE (%)                  13.00
ANNUAL INTEREST RATE FOR MORTG (%)                  12.25
CORPORATE TAX RATE (%)                              48.00
TOTAL DEPRECIATION FOR TERM OF LOAN              600000.00

PRESS ENTER TO CONTINUE ? <ENTER>

ANNUAL LEASE RENTAL                              105057.00
TOTAL LEASE PAYMENTS TO TERM                    3151720.00
TOAL LEASE TAX DEDUCTIONS                       1512820.00
TOTAL ACTUAL COST OF LEASE TO TERM              1638890.00

PRESS ENTER TO CONTINUE ? <ENTER>

ANNUAL MORTGAGE PAYMENT                          100598.00
TOTAL MORTGAGE PAYMENTS TO TERM                 3017940.00
TOTAL MORTGAGE INTEREST PAID                    2217940.00
TOTAL MORTGAGE TAXABLE EXPENSES                 2817940.00
TOTAL MORTGAGE TAX DEDUCTION                    1352610.00
TOTAL MORTGAGE COSTS AFTER TAXES                1665330.00

MORTGAGE COSTS MORE THAN LEASE BY                 26435.80
```

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

SALVAGE

To the accountant depreciation is the allocation of the cost of an asset over its useful lifetime. When considering depreciation the accountant must take into account at least the following: 1. initial cost of the asset. 2. useful lifetime of the asset and 3. salvage value or fair market value at the end of the useful life of the asset. Over the lifetime of the asset the total amount of depreciation charged is equal to the initial cost - the salvage value of the asset. At any time the BOOK VALUE of the asset is taken to equal the initial cost - total accumulated depreciation. There are many methods of allocating this amount of depreciation. Some methods like the sum of the years' digits and the declining balance method charge more depreciation per year in early years rather than in later years. One method, the straight-line method charges the same amount every year.

The declining balance method of depreciation is one method of depreciation which may be used even when the salvage value cannot be estimated directly. The declining balance method of depreciation allocation takes for depreciation in any year a fixed percentage of the previous year's book value. During the last year of the useful life of the asset enough depreciation is taken to bring the book value down to the salvage value, if known.

What **SALVAGE** does is determine what the book value of an asset being depreciated by the declining balance method will be at the end of its useful life. If the fixed rate of depreciation was carefully chosen then this book value will also be the salvage value. The program assumes this and just mentions the final salvage value.

To see how this works suppose we have an asset which will be depreciated over 10 years by the declining balance method. The yearly depreciation percentage is 16%. If the initial cost of the asset is $50000 then what salvage value is being assumed?

We run SALVAGE and after some initial documentation we see:

```
INPUT INITIAL COST IN DOLLARS ?_ We press 50000 <ENTER>

INPUT DEPRECIATION RATE(AS %) ?- We press 16 <ENTER>

INPUT NUMBER OF YEARS OF DEPRECIATION ?_ We press 10 <ENTER>


SALVAGE VALUE EQUALS 8745.08

PRESS 1 FOR MORE, 0 TO END ?_  We press 0 <ENTER>
```

SEASIND

Most businesses have seasonal variations of sales. For example, a toy store generally sells more in December than in February. How can sales in one period be compared to those in another period when those two periods normally have differences in sales? Can we predict December's sales from March's sales?

Seasonal indices are tools which help us in answering the above questions. The seasonal index for a season is simply the percentage that season's sales are of the sales for the "average" season during the year. As an example if the above toy store sells $120,000 worth of toys per year and we are talking about the four seasons then the "average" season has $30,000 sales. If in the last quarter the store sells $50,000 then the seasonal index is 167. Usually, seasonal indices are are given in percentage form.

The present program uses a simple average method of determining seasonal indices. All that is necessary is the input of sales totals for each season during several years. There may be from four to twelve "seasons" in a year and up to 100 years of data.

The following example will determine seasonal indices for widget sales. National widget has sales as in the following table:

	1975	1976	1977	1978
1st quarter	50	55	58	60
2nd quarter	35	45	43	45
3rd quarter	40	50	52	55
4th quarter	25	25	28	30

We run the program and after some initial documantation we see:

```
NUMBER OF SEASONS IN YEAR(<=12)? <4><ENTER> (We can determine indices for
                                 each of 12 months or each of 4 seasons,etc.)
NUMBER OF YEARS OF DATA AVAILABLE(<=100)? <4><ENTER>

DATA FOR YEAR  1
     ENTER DATA FOR SEASON 1 ? <50><ENTER>
     ENTER DATA FOR SEASON 2 ? <35><ENTER>
          (This input goes on until all numbers are entered. Then we see: )
AVERAGE MONTHLY SALES = $43.5

SEASON          AVERAGE SALE          INDEX

  1               $55.75              1.28
  2               $42.00               .97
  3               $49.25              1.13
  4               $27.00              0.62
```

SELLPR

A common business practice is to have an item sell for just below a break point amount, e.g. $199.95. In the olden days this was very simple. Nowadays, with the advent of sales tax the businessman must think of both his selling price and the final amount the customer pays which includes sales tax. This program will compute the final customer price given the merchant's selling price and the sales tax rate. It can be used in the following types of cases:

Someone enters the store and announces he has $200. What is the maximum selling price he can tolerate?

A merchant advertises an item at $200 including tax. What is the selling price before tax?

To use the program we RUN it and after some initial documentation we see:

```
ENTER SALES TAX RATE(%)          ? <8><ENTER>

ENTER FINAL SALE + TAX PRICE     ? <200><ENTER>

SELLING PRICE =     $185.19


INPUT 1 FOR NEW TAX RATE,2 FOR NEW CALC,SAME TAX RATE <2>


ENTER SALES TAX RATE(%)          ?8

ENTER FINAL SALE + TAX PRICE   ?<345.68><ENTER>

SELLING PRICE =      $320.07


INPUT 1 FOR NEW TAX RATE,2 FOR NEW CALC,SAME TAX RATE <1>
```

We now return to the ENTER SALES TAX RATE question after the screen is cleared.

SHARPE1

If an investor has some money to invest and a list of stocks from which to choose then a natural question to ask is what proportion of the money should be invested in each of the stocks on the list. The answer to this problem is given by modern Portfolio Theory if the investor can obtain certain data about the securities he is willing to hold and about the market in general. In particular, the expected return and the variance of return are entered are entered for the index while for each security the alpha and beta values along with variance of return are entered. A risk factor is also entered. This risk factor may range from 0 on up. Larger risk factors mean that the investor is willing to take greater risk per unit of expected return.

We shall run the following example: Using a risk factor of 1 determine the optimum proportions of the following stocks in a 3 stock portfolio:

	alpha	beta	variance
stock 1	.003	1.354	.057
stock 2	.05	-0.24	.035
stock 3	.021	0.73	.022

The market index has an expected return of .073(7.3%) and a variance of return of .011.

We run the program and after some initial documentation we see:

```
NUMBER OF STOCKS BEING CONSIDERED FOR PORTFOLIO? <3><ENTER>

ENTER 1 IF STOCK VARIANCE OF RETURN IS BEING ENTERED.
ENTER 0 IF STOCK STANDARD DEVIATION OF RETURN IS ENTERED? <1><ENTER>

ENTER STOCK 1  ALPHA,BETA,VARIANCE? <.003,1.354,.057><ENTER>
ENTER STOCK 2  ALPHA,BETA,VARIANCE? <.05,-.24,.035><ENTER>
ENTER STOCK 3  ALPHA,BETA,VARIANCE? <.021,.73,.022><ENTER>

INDEX RETURN? <.073><ENTER>
INDEX VARIANCE OF RETURN? <.011><ENTER>

ENTER RISK FACTOR ? <1><ENTER>

SOLUTION FOLLOWS

PROPORTION FOR STOCK  1  =   .360222
PROPORTION FOR STOCK  2  =   .0498532
PROPORTION FOR STOCK  3  =   .589925
PRESS ENTER FOR ANOTHER COMPUTATION ?
```

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

SIMPDISK

Discounts are common in almost all businesses at almost all times. **SIMPDISK** analyzes the following three types of situations:

(1) Given a list price and a per cent discount find the selling price
(2) given a discounted price and a per cent discount find the original price.
(3) given an original and a discounted price determine a % discount.

In this program all discount are computed relative to the original price.

Lets consider the following example: JO's DEPT STORE is selling all garden supplies at a 15% discount. Joe knows a rake usually sell for $24.95. What is the discounted price. He see the rake for $10. What was the per cent discount used by the store. Joe also sees a camera for $120 on sale. What was the original price?

We RUN **SIMPDISK** and after some initial documentation we see:

```
          MENU

1. DETERMINE SELLING PRICE GIVEN LIST PRICE & % DISCOUNT
2. DETERMINE ORIGINAL PRICE GIVEN DISCOUNTED PRICE AND % DISC.
3. DETERMINE % DISCOUNT GIVEN ORIGINAL AND DISCOUNTED PRICE

MAKE YOUR CHOICE? <1><ENTER>

SELLING PRICE GIVEN ORIGINAL PRICE AND % DISCOUNT

ORIGINAL PRICE           ? <24.95><ENTER>
% DISCOUNT               ? <15><ENTER>
DISCOUNT AMOUNT                    3.74
SELLING PRICE                     21.21

PRESS ENTER TO CONTINUE?<ENTER> - Returning to the Menu we choose option 3

DISCOUNT GIVEN ORIGINAL AND DISCOUNTED PRICES

ENTER ORIGINAL PRICE    ? <24.95><ENTER>
ENTER DISCOUNTED PRICE  ? <10><ENTER>
DISCOUNT PERCENTAGE          59.92

PRESS ENTER TO CONTINE?<ENTER> - This time we choose option 2 from Menu
ORIGINAL PRICE GIVEN SELLING PRICE AND % DISCOUNT

SELLING PRICE            ? <120><ENTER>
% DISCOUNT               ? <15><ENTER>
ORIGINAL PRICE IS           141.18
DISCOUNT AMOUNT              21.18
```

SIMPLEX

Linear programming is a mathematical tool designed to aid management make optimum use of competing resources. Problems for which linear programming may be used can be quantified as a set of linear inequalities called constraints and a linear function, called an objective function, which is to be maximized or minimized over all values of the variables for which the constraints are all true. For example, the following is a linear programming problem:

Maximize P = 120x + 75y
subject to the constraints: x >= 0, y >= 0
x+2y <= 60
4x+3y <= 120

This is a linear programming problem where P is the objective function and where there are two constraints. Note that it looks like there are four constraints but x>=0 and y>=0 are natural constraints which are not counted when numbering. This is because most of these problems involve variables which are quantities and these cannot be negative. Thus the above is a linear programming problem with 2 variables and 2 constraints.

To solve the problem above we RUN **SIMPLEX** and after some initial documentation we see:

```
NUMBER OF VARIABLES              ?<2><ENTER>
NUMBER OF CONSTRAINTS            ?<2><ENTER>
NAME OF VARIABLE 1 IS            ?<X><ENTER>
NAME OF VARIABLE 2 IS            ?<Y><ENTER>

CONSTRAINT 1
COEFFICIENT FOR VARIABLE 1   X IS ? <1><ENTER>
COFFICIENT  FOR VARIABLE 2   Y IS ? <2><ENTER>
ENTER 1 IF <= CONSTRAINT,-1 IF >= CONSTRAINT? <1><ENTER>
ENTER AMOUNT FOR CONSTRAINT  ? <60><ENTER>

CONSTRAINT 2
COEFFICIENT FOR VARIABLE 1   X IS ? <4><ENTER>
COEFFICIENT FOR VARIABLE 2   Y IS ? <3><ENTER>
ENTER 1 IF <= CONSTRAINT,-1 IF >= CONSTRAINT? <1><ENTER>
ENTER AMOUNT FOR CONSTRAINT  ? <120>

PRESS Y FOR A MINIMIZATION PROBLEM, OTHERWISE N? <N><ENTER>
ENTER COEFFICIENTS FOR OBJECTIVE FUNCTION
COEFFICIENT FOR VARIABLE 1   X IS ? <120><ENTER>
COEFFICIENT FOR VARIABLE 2   Y IS ? <75><ENTER>

VALUE OF OBJECTIVE FUNCTION = 3600
AMOUNT OF X I.E. X(1) =                30
AMOUNT OF Y I.E. X(2) =                 0
```

SINKFUND

A sinking fund is an annuity which must accumulate to some fixed amount by a given time in the future. This means that a fixed amount will be paid into a fund every period and earn interest. At the end of the last period the sum of the payments and the interest earned on the fund will just equal to the pre determined fixed amount.

In **SINKFUND** the operator inputs the future desired amount(future value), the interest rate and the number of payments, along with some information as to whether interest rates are annual or per period. The output includes the periodic payment necessary and also a table of how the sinking fund grows.

As an example suppose interest rates are 12% per year and we wish to have $5000 in our bank account at the end of 3 years. We desire to make a payment at the end of the 1st year, the end of the 2nd year and the end of the 3rd year. What is our periodic payment? We RUN **SINKFUND** and see:

```
IS INTEREST RATE QUOTED PER PERIOD(P) OR ANNUALLY(A)?<A><ENTER>

NUMBER OF PAYMENTS PER YEAR? <1><ENTER>

ENTER EACH OF THE FOLLOWING:
     FUTURE VALE              ? <5000><ENTER>
     NUMBER OF PAYMENTS       ? <3><ENTER>
     ANNUAL INTEREST RATE     ? <12><ENTER>

SINKING FUND FOR 3 PAYMENTS AT INTEREST RATE OF 12 % PER PERIOD
NUMBER OF PAYMENTS = 3
     CONSISTING OF 1 PAYMENTS ANNUALLY
ANNUAL INTEREST RATE = 12
PAYMENT             = 1481.75

PRESS ENTER TO PRINT TABLE?<ENTER>
```

END PERIOD	DEPOSIT	INT ON FUND	INCR. IN FUND	AMT IN FUND
1	1481.75	0.00	1481.75	1481.75
2	1481.75	177.81	1659.56	3141.30
3	1481.75	376.96	1858.70	5000.01
TOTALS	4445.24	554.77	5000.01	

Note the small rounding error in the final total.

SORT3

Sorting is a very common procedure used in business, play, and school. Take the example of a teacher who has a class of students and wishes to put their names in his register. First he must place these names in alphabetical order. How should he do this ? First he might put each student's name on a 3x5 card. Then he might go through the cards and try to put them in order. As a first attempt he might go through the list and look for the name that goes first in the book. After doing this he might look for the second name in the book (i.e. the first name alphabetically of those remaining after the first name is taken out). Then he might look for the third name, etc. This proocedure doesn't sound so bad but anyone who has tried it on a list of about 60 or so names will never try it again. In short, it is a very slow method of sorting.

One step faster would be to go through the list in the following way:

Take the first card out of the deck and examine the name. Examine the name on the second card. If it comes before the name on the first card we replace the second card with the first & pull the second card out. If the first card comes first alphabetically we leave the second card alone. Now compare the card we have pulled to the third card in the deck. Whichever of these two cards comes first alphabetically becomes the new pulled card. If this is the third card then this third card is pulled and is replaced by the old pulled card. Then we go to the fourth card,fifth card, etc. The result is that we have alot of swaps with names early in the alphabet tending to 'bubble' to the end of the deck. The result of the first pass through the deck is that the last name first alphabetically is definitely the last card in the deck. Now we go through the deck again using the same procedure, except ignoring the last card. If we perform this procedure a number of time about equal to the number of cards we start with we will have a sorted deck of cards. This sorting method is called the BUBBLE SORT and is also very slow. The advantage of the bubble sort is that it is very easy to understand and so is taught in alot of schools. The bubble sort is also used in several nationally sold programs as the sort of choice. If a program takes 10 minutes to sort 100 names then it probably is a bubble sort.

This program uses a sort generally faster than the bubble sort. The procedure used is called the SHELL-METZNER sort. In fact, if we sort 300 randomly arranged names then the procedure used in this program should be at least 6 times faster than a bubble sort. The program as it comes can sort alphabetic data and numeric data consisting of positive integers. Both the unsorted input list and the final sorted list may be printed on the line printer if desired.

Using the program is simple. About the only thing the user may have trouble with is the first question which asks what the maximum number of items you wish to sort is. Just pick a number greater than the most items you are thinking of sorting. After this question is answered the program sets up the needed array and we are ready to input.

STOCKVAL1

When an investor purchases shares of some stock he/she hopes for some future return, either from dividends or appreciation in the price of the stcok or both. If the investor purchases some share of a stock for $100 per share and that stock pays dividends of $6 per share then we might say that the investor is getting a return of 6% on investment. But suppose we expect dividends to increase by 10 % every year? Then next year we expect dividends of $6.60 per share and the year after that of $7.26 per share, etc. Does 6% return on investment make sense? Probably not. The model this program uses assumes that the only thing investors as a whole get out of a stock is the stream of dividends paid by the company. Though individual people make huge capital gains the only way the company itself can cause an investor to profit is through dividends. If we know how much a share of stock sells for, what the present dividend is, and how quickly we expect the dividends will grow then this program computes a long run expected rate of return on investment.

As an example of how the program works suppose we by some shares of stock in a company for $40 per share paying a dividend of $5 per share currently. We expect the dividend to increase about 10% per year. What long term rate of return can we expect on our investment?

We RUN the STOCKVAL1 program and after some documentation we see:

THE PRESENT PRICE OF 1 SHARE OF THE STOCK = ? _ (We input 40 <ENTER>)

THE PRESENT ANNUAL CASH DIVIDEND = ? _ (We input 5 <ENTER>)

EXPECTED RATE OF DIVIDEND GROWTH(IN %) = ? _ (We input 10 <ENTER>)

EXPECTED LONG RUN RATE OF RETURN = 23.75%

Note the difference between the 12.5 % nominal return and the 23.75% expected rate of return.One way this model could be used is for an investor to decide what rate of return he wants in order to purchase the stock. If this investor can estimate long run dividend rate of growth then he can use this program. If the computed value given by this program is at least as great as the investor's desired rate of return then the stock may be purchased. Otherwise do not purchase the stock.

This is but one of many stock valuation models. The user should try to understand the models available before using them in actual practice.

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

TAXDEP

Depreciation is a decrease in the value of a fixed asset due to the passage of time, wear and tear, obsolescence, or inadequacy. We record depreciation so that the cost of a fixed asset may be distributed over its useful life. On accounting statements depreciation is an expense and as such decreases earnings. However, depreciation is a special type of expense. While other expenses for the most part involve moneys actually spent during the year in question depreciation does not involve any actual flow of cash out of the corporation. It may be considered a book expense. The interesting fact is that if two different corporations have the same amount of earnings before income taxes and depreciation are considered then the corporation with more depreciation will have lower earnings but will have greater positive cash flow during the year. This means that its bank account increases more than the other corporation's. Why is this. It is because though both corporations have taken in the same amount from which taxes are to be paid, it is the corporation with greater depreciation which pays lower taxes.

TAXDEP is a program which will compute a corporation's cash flow given the pretax, predepreciation earnings and the amount of depreciation taken. The tax schedule used is the following:

Taxes owed are the following percentages of pretax earnings:

17% of the first $25000
20% of the next $25000
30% of the next $25000
40% of the next $25000
46% of all earnings above $100000

In the program NET CASH FLOW is defined as pretax,predepreciation earnings minus income taxes. We will run the program for a corporation which had $87,000 of earnings before provisions for taxes or depreciation. User input will be enclosed in < > here for illustrative purposes only. We begin and after some initial documentation we see:

```
INPUT PROFITS BEFORE DEPRECIATION ?_  <87000><ENTER>
ENTER DEPRECIATION AMOUNT AND <ENTER> BELOW
```

	DEPRECIATION	TAXES	NET CASH FLOW
?_	<10000>	$17,550.00	$69,450.00
?_	<15000>	$15,850.00	$71,150.00
?_	<25000>	$12,850.00	$74,150.00
?_	<45000>	$ 7,650.00	$79,350.00

As you see, the greater the depreciation the greater the net cash flow. The total depreciation which can be taken over the life of an asset equals no more than the initial cost of the asset so the question is when to take the depreciation. This program gives some of the consequences of taking the depreciation.

TELDIR

TELDIR sets up a directory of telephone numbers along with their owners. Each entry consists of a phone number, a person, and a comment. There is provision for using the printer to print out the entire list of names and numbers. An individual may be recalled for view or for editing. In this case the entire name must be known as this is the key used for recall. A suggestion which seems to work is to enter a name as LastName*FirstName with the * between the names. On a 16K machine this program should handle well over 200 names,although supplied dimensioning handles 200. To fit more names only the dimensioning need be changed. There is no sort presently provided so the complete list will not appear in alphabetical order but in the order of entry. There is no provision for deletions of names. However, there is provision for editing names and phone numbers. The name of a person to be deleted could be changed to zzzz, etc. or to the name of a new person.

On running the program we see, after some initial documentation:

```
          MENU

1.SET UP NEW PHONE DIRECTORY
2.ADD TO EXISTING DIRECTORY ALREADY IN MEMORY
3.READ IN A DIRECTORY ON CASSETTE
4.SAVE A DIRECTORY ON CASSETTE
5.READ IN A DIRECTORY ON DISKETTE
6.SAVE A DIRECTORY ON DISKETTE
7.REVIEW A DIRECTORY
8.MAKE CORRECTIONS
9.FIND NAME AND PHONE NUMBER

TYPE APPROPRIATE NUMBER AND ENTER?
```

All directories must be saved on diskette or cassette to be of future use. The user must name diskette directory files. Thus several separate phone directories may be stored on one diskette. If option 1 or option 2 is chosen we see the following:

```
LASTNAME*FIRSTNAME            ? <JONES*JOHN><ENTER>
A/C-PHONE # E.G.914-425-1535? <201-267-0980><ENTER>
COMMENT                       ? <FRIEND><ENTER>
```

The name is then entered into the file. Editing is about the same. The difference is that the user must enter the name exactly as in original entry to recall and edit. Finding a particular name is just like editing. For both editing and name search the program is designed to find the proper entry even if there are two identical names. The user just uses the option for wrong name which appears on the screen.

A utility sort may be added to this program to permit names to be sorted alphabetically. Then the printed list will look more orderly.

TERMSPAY

When an item is purchased the buyer sometimes has some choice regarding the method of payment. Some possibilities are that he may pay for an item in advance. If he does then sometimes there may be a cash discount. The buyer may pay COD and once again may get some type of discount. Thirdly, the item may be delivered and the buyer may have a certain amount of time to pay th elist price without penalty. Suppose the buyer is given these three choices. Which is the best way to pay? In general in problems involving different amounts of outgo at different points of time, if we wish to make a comparison, it is best to make some type of present value comparison. This is relatively simple to do in the above model. We have chosen to use the time of delovery as the present time. Then payment into the future may be discounted to the present time while payment in the past may be discounted up to the present time.

In order to use this present value analysis it is necessary for the purchaser to first estimate some interest rate for himself. This rate should be the amount he feels he needs for his money.

We show how the program works with the following problem:

A new Computer costs $1700. Delivery is in 2 months. An $85 dollar credit is given if payment accompanies the order. If payment is paid on delivery then a $10 credit is given. If payment is not made by delivery the buyer has 15 days to pay the $1700 before huge penalties are tacked on. What is the best deal if the purchaser values money at a 12% rate?

We RUN the program and after some initial documentation we see:

```
ENTER YOUR INTEREST RATE(%)? <12><ENTER>
ENTER LIST PRICE OF PURCHASE ? <1700><ENTER>
ENTER NUMBER OF MONTHS IN ADVANCE FOR EARLY PAYMENT? <2><ENTER>
ENTER DISCOUNT AMOUNT FOR MAKING EARLY PAYMENT? <85><ENTER>
ENTER DISCOUNT AMOUNT FOR COD                  ? <10><ENTER>
ENTER # MONTHS (OR PART) AFTER DELIVERY FOR NET PAYMENT? <.5><ENTER>

PRESENT VALUES ARE COMPUTED WITH RESPECT TO DAY OF DELIVERY

PRESENT VALUE OF EARLY PAYMENT = 1615
PRESENT VALUE OF PAYING COD    = 1690
PRESENT VALUE OF NET TERMS     = 1691.56

PRESS ENTER FOR MORE COMPUTATION?
```

TIMECLCK

One problem many small businesses have is adding up the time clock hours manually at the end of the week.In fact, in most cases, there isn't even any attempt to find the exact total. Every punch is rounded to the nearest 1/4 hour or the nearest 1/2 hour. This program allows the operator to enter the time puched on the time clock for time in and time out for every day of the week. The result is a summary of the hours the employee worked each day along with the weekly totals. Any weekly total above 40 hours is considered overtime and is reportec on its own line.

We shall run the ststem in an example of a two-day week: On Monday the employee arrives at 8:45 A.M., goes out to lunch at 1:00 P.M., retuns at 1:54 P.M. and leaves for the day at 5:35 P.M.. On Tuesday he arrives at 8:55 and stays until 4:37 with no lunch. When we run the system we see, afgter some initial documentation:

```
EMPLOYEE NUMBER ? <1><ENTER>

ENTER TIME FOR DAY OF WEEK GIVEN. FORM SHOULD  BE:
A OR P,HR IN.MIN IN,A OR P,HR OUT.MIN OUT
TO END DAY ENTER D,0,0,0. TO END WEEK ENTER W,0,0,0
MONDAY
A OR P,TIME IN,A OR P,TIME OUT? <A,8.45,P,1.00><ENTER>
A OR P,TIME IN,A OR P,TIME OUT? <P,1.54,P,5.35><ENTER>
A OR P,TIME IN,A OR P,TIME OUT? <D,0,0,0><ENTER>
TUESDAY
A OR P,TIME IN,A OR P,TIME OUT? <A,8.55,P,4.37><ENTER>
A OR P,TIME IN,A OR P,TIME OUT? <W,0,0,0><ENTER>


EMPLOYEE NUMBER 1
A 8.45          P     1              TOTAL =  4.25        HRS
P 1.54          P     5.35           TOTAL =  3.68333     HRS
MONDAY     TOTAL =       7.9333 HRS
A 8.55          P    4.37            TOTAL =  7.7         HRS
TUESDAY    TOTAL =       7.7    HRS
                         GRAND TOTAL = 15.63
REGULAR HOURS = 15.63          OVERTIME HOURS = 0
```

The above is printed on the line printer and the CRT. Then the machine askes for the next employee's number.

TIMEMOV

A TIME SERIES is any sequence of measurements taken on an independent variable representing time. We are always dealing with time series. Consider, for example, the following problems:

How have sales of toys varied over the year in Joe's Toy Shoppe?
How is the stock market doing?
Are interest rates getting higher or lower?

As the time series varies it generally generates a pattern which may include some long range trend. Some interesting questions which may be asked in time series analysis are the following:

Can the trend be extrapolated into the future?
Is a present value above or below the long term trend?
How is the long term trend changing over time?
Can I determine a long term trend which will 'smooth out' short term cyclical,seasonal & random variations?

The above questions all involve trend. Single observations may be extreme, e.g. one day's change in stock market prices, but when taken with other observations may be seen as part of a long term trend. A moving average type trend is gotten by replacing any single observation with the average of that observation and some previous and following observations. For example, in a 5 period moving average, the observation for the 9th period is the mean of the observstions for the 7th,8th,9th,10th & 11th period. This new average value is a so-called SMOOTHED value. A plot of smoothed values is much 'smoother' than a plot of the individual observations:

TIMEMOV computes N-period moving averages for any odd integer N. Lets show how the program works on a 3 period moving average with observations from period 1-12 as follows: 71,70,69,68,64,65,72,78,75,75,70.

We run the program and see:(USER INPUT ENCLOSED IN < >)
THIS IS AN M-PERIOD MOVING AVERAGE.WHAT DOES M EQUAL ?_ <3><ENTER>

GIVE AN UPPER BOUND ON THE NUMBER OF OBSERVATIONS YOU INTEND TO
USE. THIS WILL BE THE GREATEST NUMBER OF OBSERVATIONS POSSIBLE?_<30><ENTER>
INPUT OBSERVATIONS 1 BY 1,PRESSING ENTER AFTER EACH.
INPUT 99999 AND ENTER WHEN DONE.

OBSERVATION 1 (99999 TO END)?_ <71><ENTER>
OBSERVATION 2 (99999 TO END)?_ <70><ENTER>

We keep entering until the 12 values are in. Then we enter 99999 and we get the final result which contains smoothed values for observations 3 through 10. You will notice that when two successive observations are very different the corresponding smoothed values are not so different. Smoothing is a way of eliminating the effect of outlandish observations.

TIMETR

A TIME SERIES is any sequence of measurements taken on an independent variable representing time. We are always dealing with time series. Consider, for example, the following problems:

How have sales of toys varied over the year in Joe's Toy Shoppe?
How is the stock market doing?
Are interest rates getting higher or lower?

As the time series varies it generally generates a pattern which may include some long range trend. Some interesting questions which may be asked in time series analysis are the following:

Can the trend be extrapolated into the future?
Is a present value above or below the long term trend?
How is the long term trend changing over time?
Can I determine a long term trend which will 'smooth out' short term cyclical,seasonal & random variations?

The above questions all involve trend. Single observations may be extreme, e.g. one day's change in stock market prices, but when taken with other observations may be seen as part of a long term trend. A linear type trend is gotten by replacing any single observation with the predicted value of that observation as computed from a linear regression equation calculated from the observations on the time series. This new predicted value is a so-called SMOOTHED value. A plot of smoothed values is much 'smoother' than a plot of the individual observations. In fact in the case considered here the plot is a straight line.

TIMETR computes smoothed linear trend values for any time series. Lets show how the program works with observations from period 1-12 as follows:
71,70,73,74,73,71,76,78,75,79,80,82
We RUN the program and after some initial documentation we see:

```
GIVE AN UPPER BOUND ON THE NUMBER OF OBSERVATIONS YOU INTEND TO
USE. THIS WILL BE THE GREATEST NUMBER OF OBSERVATIONS POSSIBLE?_<30><ENTER>
INPUT OBSERVATIONS 1 BY 1,PRESSING ENTER AFTER EACH.
INPUT 99999 AND ENTER WHEN DONE.

OBSERVATION 1 (99999 TO END)?_ <71><ENTER>
OBSERVATION 2 (99999 TO END)?_ <70><ENTER>
```

We keep entering until the 12 values are in. Then we enter 99999 and we get the trend equation which is T= 68.8485 + .972027 Y where T is the trend value and Y is the year number. You will notice that when two successive observations are very different the corresponding smoothed values are not so different. Smoothing is a way of eliminating the effect of outlandish observations. In addition a table is printed giving values input, trend values=, and detrended values.

TIMUSAM

The output of **TIMUSAM** is a summary of how the subject's time is spent. To use the program the subject must categorize all activities. There is no practical limit to the number fo categories. Thus e.g. 1 may be eating, 2 may be letter writing, 3 may be commuting, etc. After entering the various categories the subject enters the different activities he has performed during the time in question. In fact for each activity in a time slot he enters the activity code an the number of minutes for the activity. When all activities are entered the program causes a printout summarizing the subject's time use habits. There is an option for line printer output though no disk or cassette data storage is offered at this time.

When we RUN the program we see some initial documentation and then

```
ENTER THE NUMBER OF DIFFERENT ACTIVITY CATEGORIES? <5><ENTER>

CATEGORY 1 DESCRIPTION(<=10 CHARACTERS)? <EATING><ENTER>
CATEGORY 2 DESCRIPTION(<=10 CHARACTERS)? <TRAVEL><ENTER>
-------
CATEGORY 5 DESCRIPTION(<=10 CHARACTERS)? <REST>
```

The categories are now displayed with an option to pront them. Then we see:

```
ENTRY 1 ENTER CATEGORY #,TIME(0,0 END   -1,0 DISPLAY)? <1,15><ENTER>
      This means 15 minutes of eating. A -1,0 would display the
      categories and then ask for the same entry number again.
ENTRY 2 ENTER CATEGORY #,TIME(0,0 END   -1,0 DISPLAY)? <2,50><ENTER>
-----------
ENTRY 24 ENTER CATEGORY #,TIME(0,0 END   -1,0 DISPLAY)? <0,0>

IS LINE PRINTER OUTPUT AVAILABLE(Y/N)?<N><ENTER>
```

CAT #	DESCRIPTION	TIME(MIN)	# OCCUR	AVG.TIME	% TOTAL
1	EATING	215.0	8	26.9	13.46
2	TRAVEL	252.0	5	50.4	15.78
3	WORK	805.0	5	161.0	50.41
4	EXERCISE	135.0	3	45.0	8.45
5	REST	190.0	2	95.0	11.90

The subject might collect time entries for a week before entering the data. It would also be useful to compare entries for different weeks.

TRANS

The transportation problem of linear programming involves selecting routes between manufacturing facilities and warehouses, or between regional warehouse and local sellers. The problem is the following: There are several sources of supply(say N) and several destinations(say M) which require those supplies. In the problem this program solves the sources of supply can supply exactly the requirements of the destinations. Thus every bit of the good is shipped from some source to some destination. The question is,"Which source will ship how much to which destination?" To help solve the problem a "transportation table must be set up". We give the number of sources and the number of destinations. Then for each source and for each destination the transportation table gives the unit cost of shipping an item from that source to that destination. The following is an example of a transportation table:

		DESTINATION			CAPACITY
		1	2	3	
	1	4	8	8	56
SOURCES	2	16	24	16	82
	3	8	16	24	77
					215
REQUIREMENT		72	102	41	215

Notice that the transportation table also gives the capacity of each source and the requirement of each destination. It is important that the sum of the capacities equals the sum of the requirements for this program.

After answering questions about the numbers of sources and the number of destinations the program leads us through entry of the transportation table as follows:

```
SOURCE 1
  UNIT COST TO DESTINATION  1 ? <4><ENTER>
  UNIT COST TO DESTINATION  2 ? <8><ENTER>
  UNIT COST TO DESTINATION  3 ? <8><ENTER>
  CAPACITY OF SOURCE 1        ? <56><ENTER>
SOURCE 2
  UNIT COST TO DESTINATION 1 ? <16><ENTER>
  UNIT COST TO DESTINATION 2 ? <24><ENTER>
  UNIT COST TO DESTINATION 3 ? <16><ENTER>
  CAPACITY OF SOURCE 2       ? <82><ENTER>
SOURCE 3
  UNIT COST TO DESTINATION 1 ? <8><ENTER>
  UNIT COST TO DESTINATION 2 ? <16><ENTER>
  UNIT COST TO DESTINATION 3 ? <24><ENTER>
  CAPACITY OF SOURCE 3       ? <77><ENTER>
```

```
REQUIREMENTS OF DESTINATION 1     ? <72><ENTER>
REQUIREMENTS OF DESTINATION 2     ? <102><ENTER>
REQUIREMENTS OF DESTINATION 3     ? <41><ENTER>
```

The computation now takes place and the solution is printed:

```
SHIPPING ASSIGNMENTS
   FROM SOURCE 1 TO DESTINATION 2    : 56       COST = 448
   FROM SOURCE 2 TO DESTINATION 1    : 41       COST = 656
   FROM SOURCE 2 TO DESTINATION 3    : 41       COST = 656
   FROM SOURCE 3 TO DESTINATION 1    : 31       COST = 248
   FROM SOURCE 3 TO DESTINATION 2    : 46       COST = 736

                  TOTAL COST                        2744
```

Even in simple problems it is difficult to see what the proper solution is. Beware about estimating the transportation table. If your cost factors are incorrect then a poor assignment may result.

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

UPSZONE

This program is a short program which will performs as a United Parcel Service Zone Chart. In particular this program emulates the zone chart for shippers in the 10901 - 10999 zip code area. To use the program all that is necessary is to RUN the program. After some initial documentation the data loads and then the user sees:

```
INPUT DESIRED ZIP CODE? <47402><ENTER>
THE UPS ZONE FOR ZIP CODE 47402 IS ZONE 5
PRESS ENTER TO CONTINUE? <ENTER>

INPUT DESIRED ZIP CODE? <34509><ENTER>
NO ZONE FOR ZIP 34509 AVAILABLE
PRESS ENTER TO CONTINUE? <ENTER>
```

Naturally not every shipper ships from the 109-- zip code area. However, the program may easily be modified by changing the DATA statements. The entries in the DATA statements are actually triples of numbers say a,b,c where a represents the three digits of the zip prefix for the lowest zip corresponding to the zone, b represents the three digits of the zip prefix for the highest zip corresponding to the zone, and c represents the zone. There are 15 entries, corresponding to 5 zones in each DATA statement. For example look at line:

```
1014 DATA 255,264,4,265,265,3,266,266,4,267,268,3,270,288,4
```

In the UPS Zone Chart we have under ZIP CODE PREFIXES the entry 255-264 and a corresponding entry of 4 under UPS ZONE. The 265,265,3 entry above corresponds to a ZIP CODE PREFIX entry having only 1 prefix.

A program like this may be used as a subroutine in a shipping program where shipping charges may automatically be computed if the UPS zone and the weight are known. One piece of advice. The DATA statements take quite alot of memory if your program makes memory a problem. One solution to this would be to store the data on a disk or cassette file and read it into the program at the beginning. This might take a little longer than using the DATA statements but will save alot of space.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

UTILITY

Utility theory has to do with one's preferences for various decision alternatives. Suppose every decision one can make leads to a known monetary payoff. Then that person will choose the decision with the highest payoff. However, if there is some uncertainty in the final payoff then the correct decision may not be so obvious. Utility theory permits us to assign a number called a 'utility to each payoff. Higher payoffs correspond to higher utilities but the relationship is not linear. Thus $10000 does not necessarily have twice the utility of $5000. Once ones' utilities are determined, utility theory indicates that the decision with the highest expected utility should be chosen.

To determine these utility numbers the payoffs are ranked from highest to lowest. The highest payoff is given some arbitrary utility. The lowest payoff is given an arbitrary lower utility. For instance, in **UTILITY** the highest utility is set equal to 10 and the lowest to 0. The utilities of the intermediate payoffs are determined in the following way: A gamble is set up. The subject is given the choice of having the intermediate payoff for sure or the top payoff with some probability p and the lowest payoff with probability 1-p. The p's are adjusted until the subject is indifferent to the sure intermediate payoff or the gamble. This p of indifference is used to compute the intermediate utility.

In the present program the subject's utilities are determined for selected monetary amouts between 0 and $100,000. By the way, different people will have different utility functions so in identical circumstances two pepole may make different decisions and yet both may be maximizing expected utility.

We RUN the program and after some initial documentation we see:

```
WITH WHAT PROBABILITY(P) WOULD YOU BE INDIFFERENT TO $ 100 FOR
SURE OR $ 100000 WITH PROBABILITY P AND 0 WITH PROBABILITY
1-P. THIS PROBABILITY MUST BE AT LEAST EQUAL TO 0

ENTER THE PROBABILITY? <.01><ENTER>

WITH WHAT PROBABILITY(P) WOULD YOU BE INDIFFERENT TO $ 1000 FOR
SURE OR $ 100000 WITH PROBABILITY P AND 0 WITH PROBABILITY
1-P. THIS PROBABILITY MUST BE AT LEAST EQUAL TO .01

ENTER THE PROBABILITY? <.15><ENTER>
```

This continues for a total of 13 preselected amounts. Then we get the following printout:

PAYOFF	UTILITY
0	0.00
100	0.10
1000	1.00

etc.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

VALADINF

A sample is a specified number of objects drawn from a much larger set of data, called the population. The purpose of drawing the sample is to attempt to something about the population. In situations in which sampling is performed it is generally acknowledged that larger samples provide more information about the population than do smaller samples. This is true in genral. However, it must be remembered that the sample should be chosen in a somewhat random manner so as to be representative of the population. A poorly chosen or measured sample, no matter how large, will not provide accurate information about the population. It is nice to be able to quantify the additional information that a sample gives. This program attempts to do such a thing in the case where the population has a normal distribution with unknown mean. We assume that we have an upper bound on the variance of the population. What we measure is something called Shannon Information for the sample distribution. This is a measure of uncertainty. We assume there is an initial sample with a certain uncertainty. Then an additional sample is taken and a new measure of uncertainty is taken. By subtracting the second uncertainty from the first we can get a measure of the change in uncertainty given by the additional information in the new sample.

As an example suppose we have a normal population in which the standard deviation is known to be no more than 10. We have already taken a sample of size 100. How much additional information(less uncertainty) will an additional sample of size 50 bring?

We RUN the program and after some initial documentation we see:

```
DO YOU HAVE BOUND FOR VARIANCE(V) OR STANDARD DEVIATION(S)? <S><ENTER>
ENTER STANDARD DEVIATION UPPER BOUND? <10><ENTER>

INITIAL SAMPLE SIZE ? <100><ENTER>
ADDITIONAL SAMPLE SIZE ? <50><ENTER>
INITIAL UNCERTAINTY WAS       2.94
FINAL UNCERTAINTY WAS         2.73727
THIS REPRESENTS A CHANGE OF .202733 UNCERTAINTY UNITS
AND IS A CHANGE OF ABOUT 6.89567%
```

It is interesting to note that if the initial sample size was 150 and if the additional sample size was still 50 then there is only a change of .143841 uncertainty units or about 5.25491%. This is an example of the law of diminishing returns. A fixed additional sample means less as the initial sample size gets larger.

VALPRINF

In the scenario to this program there are several different possible states of nature(S of N), one of which is the true state. We do not know which state is the true state but we know the probability of occurrence of each state. In addition there are several alternative actions we may take each of which yields a known amount when the state of nature is known. Since we do not know the S of N,decision theory says to choose the alternative with the highest expected value. Thus the expected value of the alternative with the highest expected value would be our long term average gain. Suppose someone offers to tell us what the S of N is. How much is this knowledge worth? This program calculates the expected value of perfect information(EVPI) or the amount the knowledge about the S of N is worth on average.

In the example we do there are 3 possible S of N's and two decisions, say a and b. The following table summarizes the information we are given:

		PAYOFFS	
S of N	Prob.	Decision a	Decision b
1	.2	$45000	$52000
2	.4	$56000	$56000
3	.4	$69000	$60000

We run the program and after some initial documantation we see:

```
ENTER NUMBER OF POSSIBLE STATES OF NATURE? <3><ENTER>

ENTER PROBABILITY OF S OF N 1 ? <.2><ENTER>
ENTER PROBABILITY OF S OF N 2 ? <.4><ENTER>
ENTER PROBABILITY OF S OF N 3 ? <.4><ENTER>

ENTER NUMBER OF POSSIBLE DECISIONS(<=8)? <2><ENTER>
DECISION 1
     PAYOFF WHEN S OF N IS 1 ? <45000><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <56000><ENTER>
     PAYOFF WHEN S OF N IS 3 ? <69000><ENTER>
DECISION 2
     PAYOFF WHEN S OF N IS 1 ? <52000><ENTER>
     PAYOFF WHEN S OF N IS 2 ? <56000><ENTER>
     PAYOFF WHEN S OF N IS 3 ? <60000><ENTER>

     DECISION            EXPECTED VALUE
        1                  59000.00
        2                  56800.00

MAXIMUM EXPECTED VALUE = 56800
EVWPI                  = 60400  (This is expected value with perfect inf.)
EXPECTED VALUE OF PERFECT INFORMATION = 3600
```

WACC

A firm may be financed in several different ways. Among these methods are the use of debt, preferred stock, common stock, and retained earnings. Each of these financing procedures has a cost (i.e. interest rate.). What this program does is have the user input the amount of financing the firm has incurred via each of the four methods of financing mentioned above along with cost of each of the types of financing. As output the program yields the weighted average cost of capital(WACC) to the firm, in a table outlining the what proportion of the firms capital is financed by each of the four methods. In terms of the cost of each of the methods of financing we have the following: the cost of common stock is based upon the sale of a new issue to the public, the cost of retained earnings is the required rate of return on equity for the company.

The following example shows how the program works:

ABC Company has $200,000,000 of capitalization split up as follows:

Debt	$ 60 million
Preferred stock	$ 20 million
Common stock	$ 40 million
Retained earnings	$ 80 million

Debt costs 4%, preferred stock 9%, new common stock 11% and retained earnings 10%.
What is the firm's WACC?

To solve we run the program and see after some initial documentation:

```
INPUT CURRENT AMOUNT OF DEBT                    ? <60><ENTER>
INPUT CAPITAL DEVOTED TO PREFERRED STOCK        ? <20><ENTER>
INPUT COMMON STOCK CAPITAL                      ? <40><ENTER>
INPUT RETAINED EARNINGS                         ? <80><ENTER>

THE FOLLOWING INPUTS ARE AFTER TAX COST

INPUT COST OF DEBT (%)                          ? <4><ENTER>
INPUT COST OF PREFERRED STOCK (%)               ? <9><ENTER>
INPUT COST OF COMMON STOCK (%)                  ? <11><ENTER>
INPUT COST OF RETAINED EARNINGS (%)             ? <10><ENTER>
```

METHOD OF FINANCING	PROPORTION (%)	COST (%)	WEIGHTED COST (%)
DEBT	30.0	4.0	1.20
PREFERRED STOCK	10.0	9.0	.90
COMMON STOCK	20.0	11.0	2.20
RETAINED EARNINGS	40.0	10.0	4.00
WEIGHTED AVERAGE COST OF CAPITAL			8.30

WARVAL

A warrant is an option to buy a stated number of shares of stock at a given price, uisually during a specified time period such as 5,10, or 20 years. The exercise price is the price which the warrant holder can pay to buy the given stock. Warrant prices are linked to such factors as exercise price, price of the underlying security, time until warrant expiration, etc. This relationship is quite complex. The theoretical(or intrinsic) value of a warrant is the present market value of the stock which may be purchased with the warrant minus the cost of exercisong the warrant. A warrant should never be priced below its intrinsic value as if it were the market professional would buy the warrant, exercise it and sell the stock for a sure profit. The present program calculates the theoretical value of a warrant. In addition, it calculates the value a warrant should actually sell at considering the above factors. A hedge ratio is also computed for warrants with several years remaining until expiration.

Input includes the following: number of shares one warrant can be used to purchase, exercise price per share of stock,market price of 1 share of stock, whether the warrant is long term(> 5 years remaining until expiration),or short term, expected annual growth rate of stocvk price, and volatility of stock price.

Output includes the theoretical and computed values of the warrant and, for short-term warrants a hedge ratio.

The following example shows how the program operates: A warrant to purchase 1 share of XYZ stock at a $40 price has 4 years until expiration. XYZ is now selling for $50 per share. Thjis price is expected to grow at a 10% annual rate. XYZ stock has a volatility of 25%. What is the value of the warrant?

We RUN the program and see:

```
HOW MANY SHARES MAY BE PURCHASED WITH ONE WARRANT?  <1><ENTER>
WHAT IS THE EXERCISE PRICE PER SHARE OF STOCK    ? <40><ENTER>
WHAT IS THE MARKET PRICE OF ONE SHARE OF STOCK   ? <50><ENTER>

IS THE WARRANT LIFE AT LEAST 5 YEARS(Y/N) ? <N><ENTER>

NUMBER OF YEARS TO EXPIRATION?  <4?><ENTER>
EXPECTED ANNUAL GROWTH RATE OF STOCK PRICE IN % ? <10><ENTER>
VOLATILITY OR ANNUAL STANDARD DEVIATION OF STOCK PR. IN % ? <25><ENTER>

THE VALUE OF THE WARRANT EQUALS   27.0531
THE HEDGE RATIO EQUALS    .694356
THE THEORETICAL VALUE OF THE WARRANT =  10
```